Swami Keshvanand Institute of Technology,
Management & Gramothan

(Accredited by NAAC with 'A^{++}' Grade)

Approved by AICTE, Ministry of Education, Government of India
Recognized by UGC under Section 2(f) of the UGC Act, 1956
Affiliated to Rajasthan Technical University, Kota

Midterm Paper, Solution, BL-CO Mapping & Attainment (Sample)

: RAMNAGARIA (JAGATPURA), JAIPUR-302017 (RAJASTHAN), INDIA
: +91-141-3500300, 2752165, 2752167, 2759609 | : 0141-2759555
: info@skit.ac.in | : www.skit.ac.in

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

II -Mid Term Examination, May-2024

Prog./Semester:	B. Tech.-VI	Branch:	CSE
Course Title:	Distributed Systems	Course Code:	6CS5-11
Duration:	1.5 Hours	Maximum Marks:	20
Session : First		Roll No.:	
Instructions if any:		Read all the questions carefully. Write neatly.	

PART -A (short answer type questions)
(All questions are compulsory)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
1	Define Byzantine Agreement Problem with its solution	2	5	1
2	Discuss failures in a DS.	2	5	2
3	Discuss how mutual exclusion is handled in DS.	2	4	2

PART -B (Analytical/Problem solving questions)
(All questions are compulsory)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
4	Illustrate Distributed Shared Memory with the help of a suitable diagram.	4	4	3
5	Illustrate fault tolerance. Explain how fault tolerance is ensured in DS.	4	5	3

PART- C (Descriptive/Analytical/Problem solving/Design questions)
(Attempt any one Question)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
6	Illustrate Distributed File System. Explain the concept of transaction control and concurrency control in detail.	6	3	3
7	Examine dynamic load sharing and load balancing and its requirement in distributed process scheduling with justification.	6	3	4

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

S. No.	Course Outcome	Percentage Coverage
CO1	List basic components of distributed systems and describe various issues of DS.	NA
CO2	Illustrate and analyze process management, Inter-process Communication techniques.	NA
CO3	Compare Distributed File Systems, and Distributed Process Scheduling with Transaction Services and Concurrency Control.	30%
CO4	Analyze Distributed Shared Memory with memory consistency models and models of distributed computation.	30%
CO5	Demonstrate different algorithms and techniques for Distributed Agreement, Replicated Data Management.	40%

S. No.	Bloom's Level	Percentage Coverage
1	BL1- Remember	14%
2	BL2- Understand	29%
3	BL3- Apply	43%
4	BL4- Analyze	14%
5	BL5- Evaluate	NA
6	BL6- Create	NA

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

A1> Mutual exclusion in distributed systems (DS) is a crucial aspect to ensure that concurrent processes or nodes do not access a shared resource simultaneously, potentially causing data corruption or inconsistencies. Unlike in single-system environments, mutual exclusion in DS has to address issues such as network latency, partial failures, and the lack of a global clock. Here are the main approaches to handle mutual exclusion in distributed systems:

1. Centralized Algorithms

Centralized Mutex Algorithm

A central coordinator node is responsible for granting access to the shared resource:

Process Request: A process sends a request to the coordinator when it wants to access the resource.

Grant Access: The coordinator grants access if no other process is currently accessing the resource; otherwise, it queues the request.

Release: The process notifies the coordinator when it releases the resource, prompting the coordinator to grant access to the next queued request.

2. Decentralized Algorithms

Token-Based Algorithms

These algorithms use a token, a unique object that grants the right to access the shared resource.

Token Ring Algorithm

Processes are arranged in a logical ring.

The token circulates around the ring.

A process can access the resource only if it holds the token. It releases the token after use, passing it to the next process in the ring.

3. Quorum-Based Algorithms

Majority Voting

Each process must request permission from a majority of the processes (a quorum).

To access the resource, a process must receive permission from a majority of processes in the quorum.

This ensures that no two processes can access the resource simultaneously, as their quorums will overlap.

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

Use consensus algorithms like Paxos or Raft to manage locks in a fault-tolerant manner.

A process requests a lock from the service, which coordinates among nodes to ensure mutual exclusion.

A2> Failures in a distributed system (DS) are common and can have significant impacts on the reliability, availability, and correctness of the system. Understanding and handling these failures are crucial for designing robust distributed systems. Here are the main types of failures that can occur in a DS and strategies for handling them:

Types of Failures

Crash Failures

Description: A node or process crashes and stops functioning entirely. The node does not respond to any messages.

Handling: Techniques like heartbeats and timeouts can detect crash failures. Redundancy and replication can help maintain availability despite such failures.

Omission Failures

Description: A node or network component fails to send or receive messages. This can be due to network issues or software bugs.

Handling: Retransmission of messages, acknowledgments, and using reliable communication protocols can mitigate omission failures.

Timing Failures

Description: Nodes or network components do not adhere to timing assumptions. For instance, a node might respond too late or too early.

Handling: Implementing timeout mechanisms and time synchronization protocols like NTP can help manage timing failures.

Response Failures

Description: A node responds incorrectly to a request. This can happen due to bugs, misconfigurations, or transient errors.

Handling: Techniques like voting, where multiple nodes provide responses and the majority decision is taken, can help detect and correct response failures.

Byzantine Failures

Description: Nodes exhibit arbitrary or malicious behavior, sending incorrect or inconsistent messages. These are the

most challenging type of failures to handle.

Handling: Byzantine Fault Tolerance (BFT) algorithms like PBFT (Practical Byzantine Fault Tolerance) and blockchain consensus mechanisms can manage Byzantine failures by ensuring agreement among honest nodes.

Causes of Failures

Hardware Failures: Physical components like servers, disks, or network hardware can fail.

Software Bugs: Bugs in the operating system, application code, or middleware can cause failures.

Network Issues: Network partitions, congestion, or misconfigurations can lead to communication failures.

Resource Exhaustion: Running out of critical resources like CPU, memory, or disk space can cause nodes to fail.

Human Errors: Misconfigurations, accidental deletions, or other operator mistakes can lead to system failures.

A3>The Byzantine Agreement Problem is a fundamental issue in distributed systems and distributed computing. It addresses the challenge of achieving consensus among distributed processes or nodes in the presence of faulty or malicious participants. The problem is named after the Byzantine generals' problem, an allegory that describes the difficulty of achieving coordinated action among generals (or nodes) when some may be traitors (or faulty/malicious nodes).

Definition of Byzantine Agreement Problem

In the Byzantine Agreement Problem, we consider a distributed system with nn nodes, some of which may be faulty or malicious (called Byzantine nodes). The goal is to ensure that all non-faulty nodes agree on a single value despite the presence of Byzantine nodes. The requirements for a solution to the Byzantine Agreement Problem are:

Agreement: All non-faulty nodes must agree on the same value.

Validity: If all non-faulty nodes propose the same value, they must agree on that value.

Termination: All non-faulty nodes must eventually reach a decision.

Solution to the Byzantine Agreement Problem

One of the well-known solutions to the Byzantine Agreement Problem is the Byzantine Fault Tolerance (BFT) algorithm. The most notable BFT algorithm is the Practical Byzantine Fault Tolerance (PBFT), proposed by Miguel Castro and Barbara Liskov in 1999. Here's an overview of the PBFT algorithm:

PBFT Algorithm

The PBFT algorithm is designed for asynchronous systems and can tolerate up to $\lfloor n-13 \rfloor \lfloor 3n-1 \rfloor$ Byzantine faults,

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

meaning it can handle ff faulty nodes out of nn total nodes, where $n \geq 3f+1$n≥3f+1.

Phases of PBFT

Pre-prepare Phase:

A client sends a request to the primary node (leader).

The primary node multicasts a pre-prepare message to all replica nodes.

Prepare Phase:

Upon receiving the pre-prepare message, each replica node multicasts a prepare message to all other nodes.

Each node waits to receive $2f$2f prepare messages (including its own).

Commit Phase:

After receiving $2f$2f prepare messages, a node multicasts a commit message to all other nodes.

Each node waits to receive $2f$2f commit messages (including its own).

Reply Phase:

After receiving $2f$2f commit messages, the nodes execute the request and send a reply to the client.

The client waits for $f+1$f+1 consistent replies to consider the request complete.

Key Features

Fault Tolerance: The PBFT algorithm can tolerate up to $\lfloor n-13 \rfloor \lfloor 3n-1 \rfloor$ Byzantine faults, ensuring consensus despite malicious behavior.

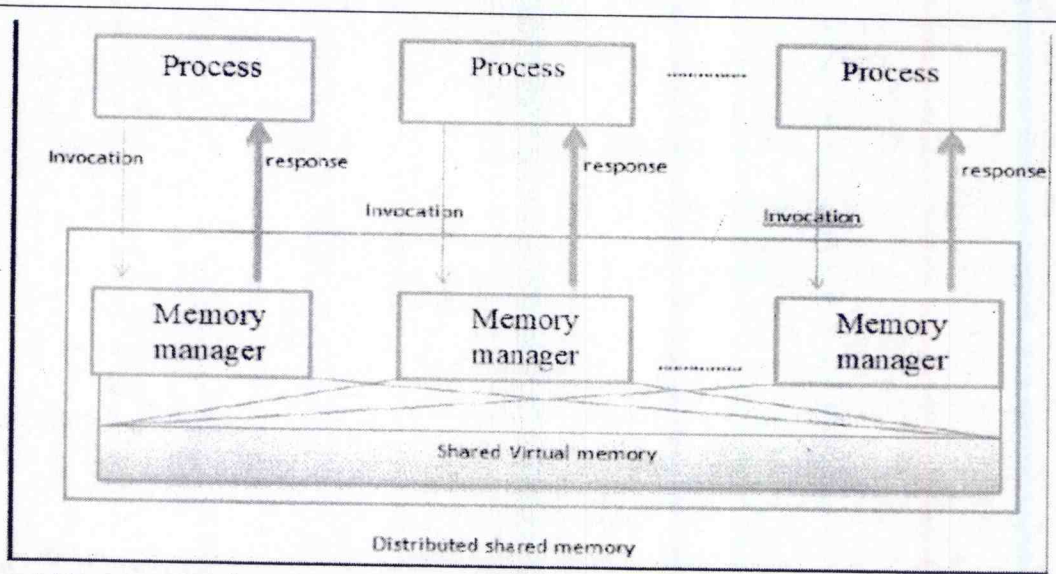
Safety and Liveness: The algorithm guarantees safety (agreement and validity) and liveness (termination) even in asynchronous environments.

Low Latency: PBFT achieves consensus with a lower message complexity and latency compared to earlier Byzantine agreement protocols.

A4>

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		



Distributed Shared Memory (DSM) is an abstraction that allows distributed systems to share a single memory space among multiple nodes. This abstraction enables processes on different machines to access shared data as if they were on the same physical memory. DSM simplifies the development of distributed applications by providing a shared memory model, abstracting away the complexities of message passing.

Key Concepts of DSM

Shared Memory Abstraction:

Provides a unified memory space accessible by all nodes in the system.

Allows processes to read and write to shared memory locations transparently.

Consistency Models:

Strict Consistency: Every read returns the most recent write (impractical in most DSM systems due to high latency and synchronization overhead).

Sequential Consistency: Operations appear in some sequential order that is consistent with the order observed by each process.

Causal Consistency: Ensures that causally related operations are seen by all nodes in the same order.

Eventual Consistency: Updates will eventually propagate to all nodes, and all nodes will converge to the same value if no new updates are made.

Memory Coherence:

Ensures that all nodes have a consistent view of the shared memory.

Typically implemented using coherence protocols such as invalidation or update protocols.

Diagram of DSM

Each node has its own local memory.

Nodes are connected to the shared memory space provided by DSM.

Distributed Shared Memory:

Acts as a shared space accessible by all nodes.

Manages shared pages that can be mapped into the local memory of each node.

Shared Pages:

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

Segments of memory that are part of the DSM.
Each shared page can be accessed and modified by any node in the system.
Working of DSM
Memory Mapping:
The DSM system maps shared pages into the local memory of each node.
When a process accesses a shared memory location, the DSM ensures that the latest version of the page is available.
Synchronization:
DSM uses synchronization mechanisms like locks or semaphores to manage access to shared data.
Ensures consistency and prevents race conditions.
Consistency Protocols:
Invalidation Protocol: When a node modifies a shared page, it invalidates copies of that page on other nodes.
Update Protocol: When a node modifies a shared page, it updates the copies of that page on other nodes.
Fault Tolerance:
DSM systems often include mechanisms to handle node failures, ensuring data replication and recovery.
Use Cases of DSM
Parallel Computing:
DSM is used in parallel computing environments where multiple processors need to access shared data efficiently.
Distributed Databases:
Provides a shared memory space for distributed database systems, facilitating coordinated access to database records.
Collaborative Applications:
Enables multiple users to work on shared documents or resources in real-time, with consistent views of the shared data.

A5>Definition of Fault Tolerance

Fault tolerance is the property of a system that enables it to continue operating properly in the event of the failure of some of its components. A fault-tolerant system is designed to ensure that the system as a whole remains functional and maintains its performance and availability, even when certain parts of the system fail. Fault tolerance is crucial in distributed systems (DS) due to the inherent complexity and higher likelihood of component failures.

Ensuring Fault Tolerance in Distributed Systems

Fault tolerance in distributed systems is achieved through a combination of techniques designed to detect, isolate, and recover from failures. Here are the key strategies and mechanisms used to ensure fault tolerance in distributed systems:

1. Redundancy

Description: Redundancy involves duplicating critical components or data to ensure that if one component fails, another can take over without interrupting the system's functionality.

Data Redundancy: Data is replicated across multiple nodes to prevent data loss and ensure data availability. Examples

include RAID (Redundant Array of Independent Disks) and distributed databases like Cassandra, which replicate data across multiple nodes.

Component Redundancy: Critical system components, such as servers or network links, are duplicated. If one component fails, the redundant component takes over. This is common in high-availability setups.

2. Replication

Description: Replication involves creating copies of data or services and distributing them across multiple nodes or locations to ensure fault tolerance.

Master-Slave Replication: One node acts as the master (primary), and other nodes act as slaves (secondaries). If the master fails, a slave can be promoted to master.

Multi-Master Replication: Multiple nodes act as masters, allowing read and write operations. This ensures that even if one master fails, others can continue to serve requests.

3. Consensus Algorithms

Description: Consensus algorithms are used to achieve agreement among distributed nodes on a single data value or system state, even in the presence of some faulty nodes.

Paxos and Raft: These are consensus protocols used to ensure that multiple nodes agree on a single value or sequence of operations. They are commonly used in distributed databases and systems like etcd and Consul.

Byzantine Fault Tolerance (BFT): Algorithms like PBFT (Practical Byzantine Fault Tolerance) handle Byzantine failures, where nodes may behave arbitrarily or maliciously. These are used in blockchain systems and other critical applications.

4. Failure Detection

Description: Failure detection mechanisms identify when components or nodes have failed.

Heartbeats: Nodes periodically send heartbeat messages to indicate that they are alive. If a heartbeat is missed for a certain period, the node is considered to have failed.

Timeouts: If a node does not respond to a request within a specified time, it is assumed to have failed.

5. Failover Mechanisms

Description: Failover mechanisms automatically switch operations to a standby system or component when a primary system or component fails.

Active-Passive Failover: A standby system (passive) takes over when the primary system (active) fails. This is common in database clusters and web server setups.

Active-Active Failover: All systems are active and share the load. If one system fails, the remaining systems continue to

handle the workload. This is used in load-balanced server farms.

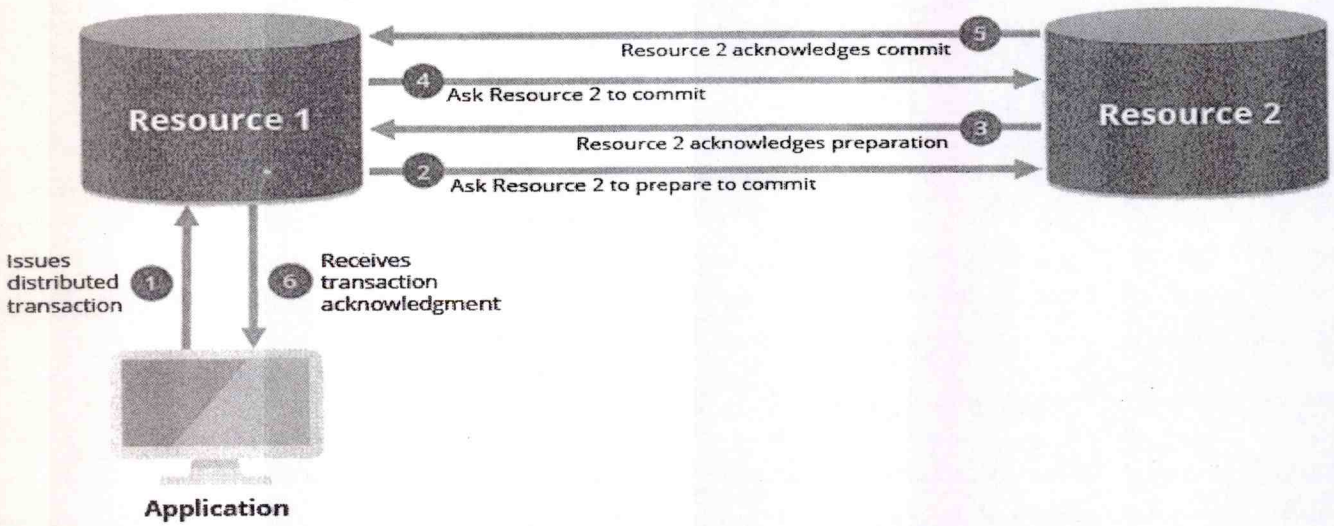
A6> Distributed File System (DFS)

A Distributed File System (DFS) is a system that allows files to be accessed and managed across multiple servers in a network, presenting a unified file system view to users and applications. The goal of a DFS is to provide reliable, efficient, and transparent access to files regardless of their physical location in the network.

A distributed transaction is a set of operations on data that is performed across two or more data repositories (especially databases). It is typically coordinated across separate nodes connected by a network, but may also span multiple databases on a single server.

There are two possible outcomes: 1) all operations successfully complete, or 2) none of the operations are performed at all due to a failure somewhere in the system. In the latter case, if some work was completed prior to the failure, that work will be reversed to ensure no net work was done. This type of operation is in compliance with the "ACID" (atomicity-consistency-isolation-durability) principles of databases that ensure data integrity. ACID is most commonly associated with transactions on a single database server, but distributed transactions extend that guarantee across multiple databases.

The operation known as a "two-phase commit" (2PC) is a form of a distributed transaction. "XA transactions" are transactions using the XA protocol, which is one implementation of a two-phase commit operation.



A distributed transaction spans multiple databases and guarantees data integrity.

How Do Distributed Transactions Work?

Distributed transactions have the same processing completion requirements as regular database transactions, but they must be managed across multiple resources, making them more challenging to implement for database developers. The multiple resources add more points of failure, such as the separate software systems that run the resources (e.g., the

database software), the extra hardware servers, and network failures. This makes distributed transactions susceptible to failures, which is why safeguards must be put in place to retain data integrity.

For a distributed transaction to occur, transaction managers coordinate the resources (either multiple databases or multiple nodes of a single database). The transaction manager can be one of the data repositories that will be updated as part of the transaction, or it can be a completely independent separate resource that is only responsible for coordination. The transaction manager decides whether to commit a successful transaction or rollback an unsuccessful transaction, the latter of which leaves the database unchanged.

First, an application requests the distributed transaction to the transaction manager. The transaction manager then branches to each resource, which will have its own "resource manager" to help it participate in distributed transactions. Distributed transactions are often done in two phases to safeguard against partial updates that might occur when a failure is encountered. The first phase involves acknowledging an intent to commit, or a "prepare-to-commit" phase. After all resources acknowledge, they are then asked to run a final commit, and then the transaction is completed.

We can examine a basic example of what happens when a failure occurs during a distributed transaction. Let's say one or more of the resources become unavailable during the prepare-to-commit phase. When the request times out, the transaction manager tells each resource to delete the prepare-to-commit status, and all data will be reset to its original state. If instead, any of the resources become unavailable during the commit phase, then the transaction manager will tell the other resources that successfully committed their portion of the transaction to undo or "rollback" that transaction, and once again, the data is back to its original state. It is then up to the application to retry the transaction to make sure it gets completed.

Why Do You Need Distributed Transactions?

Distributed transactions are necessary when you need to quickly update related data that is spread across multiple databases. For example, if you have multiple systems that track customer information and you need to make a universal update (like updating the mailing address) across all records, a distributed transaction will ensure that all records get updated. And if a failure occurs, the data is reset to its original state, and it is up to the originating application to resubmit the transaction.

Concurrency Control in Distributed Transactions

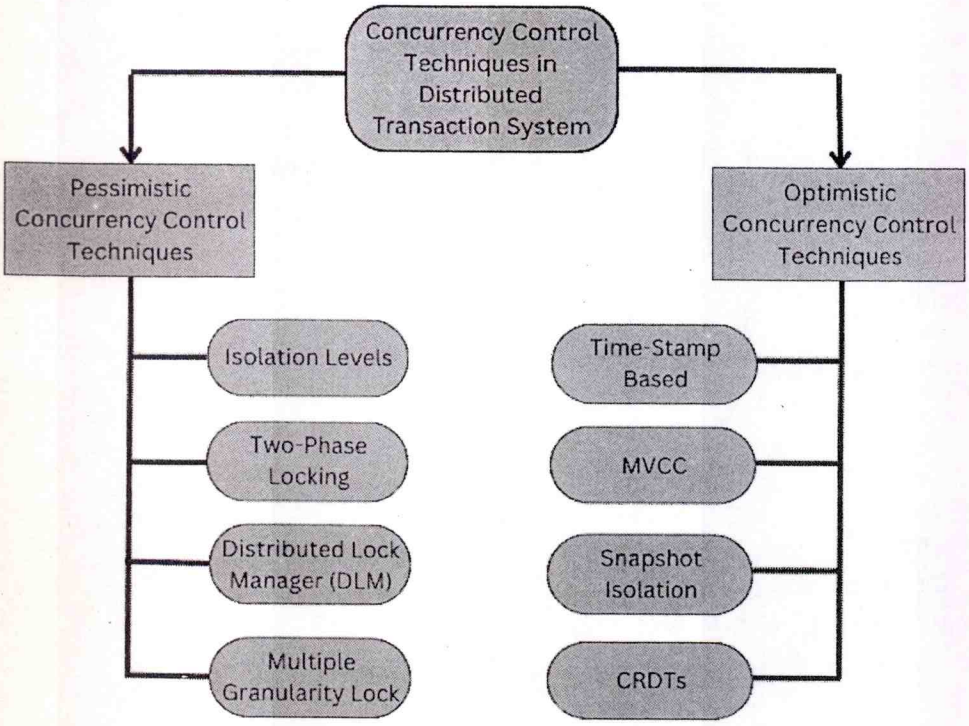
Concurrency control mechanisms provide us with various concepts & implementations to ensure the execution of any transaction across any node doesn't violate ACID or BASE (depending on database) properties causing inconsistency & mixup of data in the distributed systems. Transactions in the distributed system are executed in "sets", every set consists of various sub-transactions. These sub-transactions across every node must be executed serially to maintain data integrity & the concurrency control mechanisms do this serial execution.

Types of Concurrency Control Mechanisms

There are 2 types of concurrency control mechanisms as shown below diagram:

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		



Types of Concurrency Control Mechanism

Pessimistic Concurrency Control (PCC)

The Pessimistic Concurrency Control Mechanisms proceeds on assumption that, most of the transactions will try to access the same resource simultaneously. It's basically used to prevent concurrent access to a shared resource and provide a system of acquiring a Lock on the data item before performing any operation.

Optimistic Concurrency Control (OCC)

The problem with pessimistic concurrency control systems is that, if a transaction acquires a lock on a resource so that no other transactions can access it. This will result in reducing concurrency of the overall system.

The Optimistic Concurrency control techniques proceeds on the basis of assumption that, 0 or very few transactions will try to access a certain resource simultaneously. We can describe a system as FULLY OPTIMISTIC, if it uses No-Locks at all & checks for conflicts at commit time. It has following 4-phases of operation:

Read Phase: When a transaction begins, it reads the data while also logging the time-stamp at which data is read to verify for conflicts during the validation phase.

Execution Phase: In this phase, the transaction executes all its operation like create, read, update or delete etc.

Validation Phase: Before committing a transaction, a validation check is performed to ensure consistency by checking

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

the last_updated timestamp with the one recorded at read_phase. If the timestamp matches, then the transaction will be allowed to be committed and hence proceed with the commit phase.

Commit phase: During this phase, the transactions will either be committed or aborted, depending on the validation check performed during previous phase. If the timestamp matches, then transactions are committed else they're aborted.

Pessimistic Concurrency Control Methods

Following are the four Pessimistic Concurrency Control Methods:

Isolation Level

The isolation levels are defined as a degree to which the data residing in Database must be isolated by transactions for modification. Because, if some transactions are operating on some data let's say transaction – T1 & there comes another transaction – T2 and modifies it further while it was under operation by transaction T1 this will cause unwanted inconsistency problems. Methods provided in this are: Read-Uncomitted, Read-Comitted, Repeatable Read & Serializable.

Two-Phase Locking Protocol

The two-phase locking protocol is a concurrency technique used to manage locks on data items in database. This technique consists of 2 phases:

Growing Phase: The transaction acquires all the locks on the data items that'll be required to execute the transaction successfully. No locks will be realease in this phase.

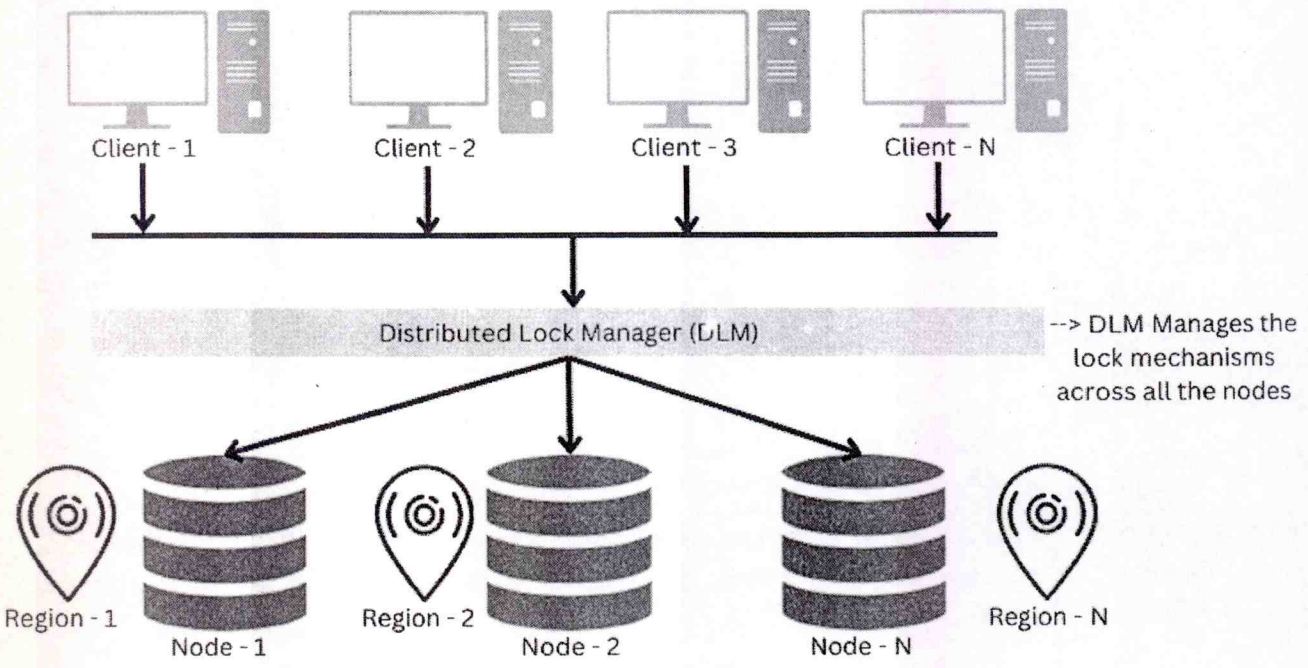
Shrinking Phase: All the locks acquired in previous phase will be released one by one and No New locks will be acquired in this phase.

Distributed Lock Manager

A distributed lock a critical component in the distributed transaction system, which co-ordinates the lock acquiring, and releasing operations in the transactions. It helps in synchronizing the transaction and their operation so that data integrity is maintained.

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		



Distributed Lock Manager (DLM)

Multiple Granularity Lock

A lock can be acquired at various granular level like: table level, row/record level, page level or any other resource's level. In transaction system a transaction can lock a whole table, or a specific row while performing some changes on it. This lock acquiring when done by various transactions simultaneously, this phenomena is called as multiple granularity locking.

Optimistic Concurrency Control Methods

Below are four Optimistic Concurrency Control Methods:

Timestamp Based (OCC)

In a timestamp based concurrency technique, each transaction in the system is assigned a unique timestamp which is taken as soon as the transaction begins, and its verified again during the commit phase. If there's new updated timestamp from a different transaction then based on some policy defined by the System Adminstrator the transaction will either be restarted or aborted. But if the times stamp is same & never modified by any other transaction then it will be committed.

Example: Let's say we have two transaction T1 and T2, they operate on data item – A. The Timestamp concurrency technique will keep track of the timestamp when the data was accessed by transaction T1 first time.

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

Transaction	Data item and operation	Most_recent_Timestamp	Initial_timestamp of data item (A)
T1	Read(A)	12:00PM	12:00PM
T2	Write(A)	12:15PM	12:00PM
T1	Write(A)	12:30PM	12:00PM

Now, let's say this transaction T1 is about to commit, before committing, it will check the initial timestamp with the most recent timestamp. In our case, the transaction T1 won't be committed because a write operations by transaction T2 was performed.

if(Initial_timestamp == Most_recent_timestamp)
then 'Commit'
else
'Abort'

In our case, transaction will be aborted because T2 modified the same data item at 12:15PM.

Multi-Version Concurrency Control (MVCC)

In MVCC, every data item has multiple versions of itself. When a transaction starts, it reads the version that is valid at the start of the transaction. And when the transaction writes, it creates a new version of that specific data item. That way, every transaction can concurrently perform their operations.

Example: In a banking system two or more user can transfer money without blocking each other simultaneously.

A similar technique to this is : Immutable Data Structures. Every time a transaction performs a new operation, new data item will be created so that way transactions do not have to worry about consistency issues.

Snapshot Isolation

Snapshot isolation is basically a snapshot stored in an isolated manner when our database system was purely consistent. And this snapshot is read by the transactions at the beginning. Transaction ensures that the data item is not changed while it was executing operations on it. Snapshot isolation is achieved through OCC & MVCC techniques.

Conflict Free Replicated Data Types (CRDTs)

CRDTs is a data structure technique which allows a transaction to perform all its operation and replicate the data to

some other node or current node. After all the operations are performed, this technique offers us with merging methods that allows us to merge the data across distributed nodes (conflict-free) and eventually achieving consistent state (eventually consistent property).

A7> Dynamic Load Sharing and Load Balancing in Distributed Systems

In distributed systems, load sharing and load balancing are critical for ensuring that computational resources are utilized efficiently and that no single node is overwhelmed while others remain idle. These concepts play a key role in distributed process scheduling.

Load Sharing

Load Sharing refers to the distribution of tasks across multiple nodes in a distributed system to ensure that all nodes have a roughly equal amount of work. It does not necessarily aim to minimize the overall completion time of tasks but focuses on preventing any node from being idle while others are overloaded.

Dynamic Load Sharing involves making real-time decisions about where to allocate tasks based on the current load of each node. This is in contrast to static load sharing, where the decision is made at the start based on predefined criteria.

Key Features of Dynamic Load Sharing:

Real-Time Decision Making: Tasks are assigned to nodes based on their current load, which can change dynamically.

Adaptability: The system adapts to changing workloads and node availability.

Simple Mechanisms: Common methods include random allocation, round-robin, and threshold-based policies.

Load Balancing

Load Balancing aims to distribute tasks across nodes to minimize the overall completion time, maximize resource utilization, and improve the system's performance. It is more sophisticated than load sharing and often includes mechanisms to monitor node loads continuously and migrate tasks as necessary.

Dynamic Load Balancing involves real-time monitoring and adjustment of task distribution to ensure an even workload across all nodes.

Key Features of Dynamic Load Balancing:

Continuous Monitoring: Constantly monitors the load on each node.

Task Migration: Tasks can be moved between nodes to balance the load.

Complex Algorithms: Often employs more sophisticated algorithms like least connection, shortest expected delay, or weighted round-robin.

Requirement in Distributed Process Scheduling

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

Distributed process scheduling is the task of deciding where to execute processes in a distributed system. Both dynamic load sharing and load balancing are essential for effective distributed process scheduling for several reasons:

Improved Performance:

By evenly distributing tasks, the system can ensure that all nodes are effectively utilized, reducing the overall completion time for processes.

Prevents bottlenecks where one node might become a performance limiter due to excessive load.

Scalability:

As the system grows, dynamic load balancing ensures that new nodes can be integrated seamlessly without overloading existing nodes.

Adapts to varying workloads, which is crucial for large-scale distributed systems.

Fault Tolerance and Reliability:

Dynamic load balancing can detect node failures and redistribute tasks from the failed node to other nodes, maintaining system reliability.

Enhances the system's ability to handle failures without significant performance degradation.

Resource Utilization:

Maximizes resource utilization by ensuring that all nodes are active and contributing to the computational workload.

Avoids scenarios where some nodes are idle while others are overloaded, leading to inefficient use of resources.

Energy Efficiency:

By distributing tasks more evenly, the system can prevent nodes from overheating and reduce energy consumption.

Dynamic strategies can adapt to energy-efficient policies, such as turning off idle nodes or reducing their power usage.

Justification

Why Dynamic Load Sharing and Balancing are Necessary:

Heterogeneous Environments: In many distributed systems, nodes can have different capacities and performance characteristics. Dynamic load balancing can take these differences into account, ensuring that more capable nodes handle larger workloads.

Variable Workloads: Workloads in distributed systems can be highly variable and unpredictable. Static load balancing cannot adapt to these changes, whereas dynamic load balancing can adjust in real-time to current conditions.

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B.Tech. VI Sem	Course Title: Distributed Systems	Course Code:6CS5-11
Duration: 1.5 hours	Date: 31/05/2024 Session: FIRST	Max Marks: 20
Submitted By: Dr. Nilam Choudhary/Ms. Astha Joshi		

User Experience: For systems that provide services to users (e.g., web servers, cloud services), maintaining a balanced load ensures faster response times and a better user experience.

Cost Efficiency: Efficiently utilizing resources can reduce operational costs, particularly in cloud environments where resources are billed based on usage.

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

II -Mid Term Examination, May-2024

Prog./Semester:	**B. Tech.-VI**	**Branch:**	**CSE(AI)/ IT**
Course Title:	**DIP**	**Course Code:**	**6CAI3-01/6IT3-01**
Duration:	**1.5 Hours**	**Maximum Marks:**	**20**
Session : First		**Roll No.:**	
Instructions if any:			

PART -A (short answer type questions)
(All questions are compulsory)

Q. No.	Question	Max. Marks	CO (s)	Bloom's Level
1	What are the common causes of image degradation in digital images?	2	3	1
2	What are the most common edge detection algorithms used in image processing?	2	5	1
3	Explain the term 'redundancy'? If an image is 10 MB before compression and 5 MB after compression, Find the compression ratio and saving percentage.	2	4	2

PART -B (Analytical/Problem solving questions)
(All questions are compulsory)

Q. No.	Question	Max. Marks	CO (s)	Bloom's Level
4	Compare the various types of noises in digital images with the help of suitable graphs.	4	3	4
5	Analyze the hough transform to find out whether the following points (1, 4), (2, 3), (3, 1), (4, 1), (5, 0) are collinear or not. Also find the equation of line.	4	5	4

PTO

PART- C (Descriptive/Analytical/Problem solving/Design questions)

(Attempt any one Question)

<table>
<tr><th>Q. No.</th><th>Question</th><th>Max. Marks</th><th>CO (s)</th><th>Bloom s Leve</th></tr>
<tr><td>6</td><td>Consider the probabilities p(a)=0.2, p(b)=0.3, p(c)=0.1, p(d)=0.4.Encode message "abcad", comprising a string by apply arithmetic coding to compress the message?</td><td>6</td><td>4</td><td>3</td></tr>
<tr><td>7</td><td>Determine the Huffman code assignment procedure for the following Data.
<table>
<tr><td>Symbol</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>Probability</td><td>0.1</td><td>0.4</td><td>0.06</td><td>0.1</td><td>0.04</td><td>0.3</td></tr>
</table></td><td>6</td><td>4</td><td>5</td></tr>
</table>

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

S. No.	Course Outcome	Percentage Coverage
CO1	**Identify** the fundamental elements of an Image and **Describe** the need of Digital Image Processing	0%
CO2	**Understand** different types of image transformation techniques and their properties.	0%
CO3	**Use** various noise models and **Calculate** the values for restoration and degradation.	23%
CO4	**Analyze** and **Evaluate** various image compression techniques.	54%
CO5	**Integrate** and **Demonstrate** various image transformation and segmentation techniques.	23%

S. No.	Bloom's Level	Percentage Coverage
1	BL1- Remember	15%
2	BL2- Understand	8%
3	BL3- Apply	23%
4	BL4- Analyze	31%
5	BL5- Evaluate	23%
6	BL6- Create	0

-- Question Paper Solution --

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	**Course Title: DIP**	**Course Code: 6CAI3-01/6IT3-01**
Duration: 1.5 hours	**Date: 28.05.2024** **Session : FIRST**	**Max Marks: 20**
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Q.1 What are the common causes of image degradation in digital images?

Solution: Image degradation in digital images can be caused by various factors, including:

1. Noise: This includes random variations in brightness or color in an image. Noise can be introduced during image capture, transmission, or processing. Common types of noise are:

Gaussian Noise: A statistical noise that has a probability density function equal to that of the normal distribution.

Salt-and-Pepper Noise: Occurs when pixels in an image are either very bright or very dark, giving the appearance of white and black dots.

2. Blur: This occurs due to the loss of sharpness and detail in an image and can result from:

Motion Blur: Caused by the movement of the camera or subject during exposure.

Out-of-Focus Blur: Occurs when the camera lens is not focused correctly.

Atmospheric Blur: Caused by particles or turbulence in the air affecting the clarity of the captured image.

3. Compression Artifacts: Distortions that occur when an image is compressed, especially using lossy compression algorithms like JPEG. These artifacts include:

4. Resolution Loss: This results from reducing the image size or using a lower resolution sensor during capture, leading to a loss of detail and clarity.

5. Color Distortion: Changes in the original colors of an image due to:

6. Digital Sampling: Errors introduced during the conversion of an analog image to a digital format, such as:

7. Quantization Errors: Loss of image detail resulting from reducing the number of bits used to represent each pixel's color value during compression or processing.

8. Environmental Factors: External conditions like dust, humidity, and temperature variations can affect the image sensor and degrade image quality.

Q.2 What are the most common edge detection algorithms used in image processing?

Solution: Edge detection is a critical step in image processing and computer vision, used to identify the boundaries of objects within images. Here are some of the most common edge detection algorithms:

Sobel Operator: Uses two 3x3 convolution kernels, one for detecting changes in the horizontal direction and one for the vertical direction. Highlights regions of high spatial frequency, typically corresponding to edges.

Prewitt Operator: Similar to the Sobel operator, but uses different kernels for gradient approximation.Often used in edge detection to approximate the gradient of the image intensity function.

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Canny Edge Detector: A multi-stage algorithm that involves smoothing the image with a Gaussian filter, finding intensity gradients, applying non-maximum suppression, and using double thresholding and edge tracking by hysteresis. Known for its effectiveness in detecting true edges and reducing noise.

Laplacian of Gaussian (LoG): Applies a Gaussian smoothing filter to the image and then uses the Laplacian operator to detect edges. Detects edges by looking for zero-crossings after the Laplacian filter is applied.

Roberts Cross Operator: Uses 2x2 convolution kernels to approximate the gradient of the image intensity function. One of the oldest edge detection algorithms, simple but sensitive to noise.

Scharr Operator: An improvement over the Sobel operator, providing better rotational symmetry and accuracy. Often preferred when high precision in edge detection is required.

Kirsch Operator: Uses eight convolution kernels to detect edges in all eight compass directions (N, NE, E, SE, S, SW, W, NW). Provides directional edge detection and can be useful in specific applications where directionality is important.

Marr-Hildreth (Laplacian of Gaussian): Combines Gaussian smoothing with the Laplacian operator to find edges, similar to LoG. Effective in detecting edges and reducing noise, although it can sometimes produce false edges due to the second derivative.

Q.3 Explain the term 'redundancy'? If an image is 10 MB before compression and 5 MB after compression, Find the compression ratio and saving percentage.

Solution: Redundancy in the context of information theory and image processing refers to the presence of unnecessary or repetitive information that can be removed or compressed without significantly affecting the quality or understanding of the data. It is a key concept in compression algorithms, which aim to reduce file sizes by eliminating this redundant information.

Types of Redundancy

Spatial Redundancy, Temporal Redundancy, Spectral Redundancy, Psycho-visual Redundancy

Compression Ratio=10/5 =2MB,
Percentage Reduction=50%

Q.4 Compare the various types of noises in digital images with the help of suitable graphs.

Solution: In digital images, various types of noise can degrade image quality. Each type of noise has distinct characteristics and impacts image processing differently. Below, I compare the common types of noise in digital images with descriptions and graphical representations:

1. Gaussian Noise: Gaussian noise is statistical noise with a probability density function equal to that of the normal distribution (Gaussian distribution). It is characterized by its mean (average intensity) and variance (spread).

Graphical Representation: A histogram of Gaussian noise typically shows a bell-shaped curve centered around the mean value.

2. Salt-and-Pepper Noise: Salt-and-pepper noise, also known as impulse noise, appears as sparsely occurring white and black pixels in an image. It is caused by sudden disturbances in the image signal.

Graphical Representation: The noise histogram shows two distinct peaks, one near the minimum value (black) and one near the maximum value (white), with the original data in between.

3. Poisson Noise (Shot Noise): Poisson noise, also known as shot noise, is related to the discrete nature of electronic charge. It is commonly seen in photon-counting devices like digital cameras.
Graphical Representation: The histogram for Poisson noise resembles the Poisson distribution, which is skewed and has a single peak.
4. Speckle Noise: Speckle noise is multiplicative noise commonly seen in radar and medical imaging. It appears as a granular pattern.
Graphical Representation: The noise histogram shows a multiplicative effect where noise is proportional to the intensity of the image pixels.
5. Uniform Noise: Uniform noise has a constant probability distribution, meaning each value in the range has an equal probability of occurring. It is often used to model quantization noise.
Graphical Representation: The histogram for uniform noise is flat, indicating that all values are equally probable.

Summary of Characteristics

Noise Type	Characteristics	Histogram Shape
Gaussian Noise	Bell-shaped distribution, mean and variance	Bell-shaped curve
Salt-and-Pepper	Sparse black and white pixels	Two peaks at extremes (black and white)
Poisson Noise	Related to signal intensity, skewed	Poisson distribution
Speckle Noise	Multiplicative noise, grainy appearance	Dependent on image intensity
Uniform Noise	Equal probability of all values	Flat distribution

Conclusion: Understanding the types of noise and their characteristics is essential for selecting appropriate noise reduction techniques in image processing. Each noise type requires different strategies for effective removal or reduction, thus preserving the quality and integrity of the digital image.

Swami Keshvanand Institute of Technology, Management & Gramothan,
Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Q.5 Analyze the hough transform to find out whether the following points (1, 4), (2, 3), (3, 1), (4, 1), (5, 0) are collinear or not. Also find the equation of line.

Solution:

Solution. The equation of line is $y = mx + c$

- In order to perform Hough transform, we need to convert line from (x, y) plane to (m, c) plane.
- equation of line in (m, c) plane is $\boxed{c = -mx + y}$

⇒ For $(x, y) = (1, 1)$

$c = -m + 1$

$\left\{\begin{array}{ll} \text{if } c = 0 & \text{then } m = 1 \\ \text{if } m = 0 & \text{then } c = 1 \end{array}\right\}$ So

Thus $(m, c) = (1, 1)$

⇒ For $(x, y) = (2, 2)$ $\boxed{c = -mx + y}$

$c = -2m + 2$

$\left\{\begin{array}{ll} \text{if } c = 0 & \text{then } m = 1 \\ \text{if } m = 0 & \text{then } c = 2 \end{array}\right\}$

Thus $(m, c) = (1, 2)$

⇒ For $(x, y) = (3, 3)$ $c = -3m + 3$

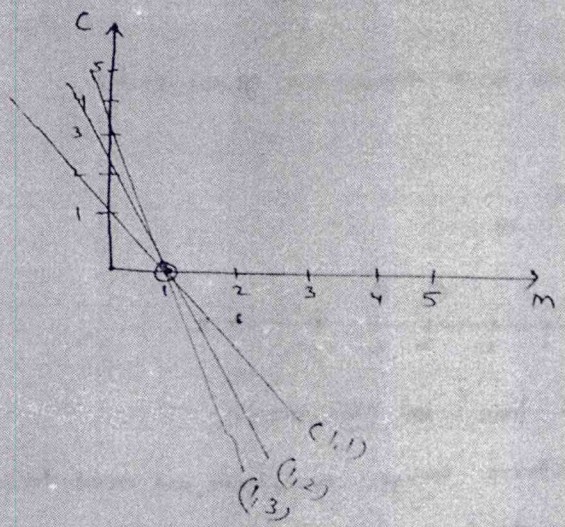
$\left\{\begin{array}{ll} \text{if } c = 0 & \text{then } m = 1 \\ \text{if } m = 0 & \text{then } c = 3 \end{array}\right\}$

Thus $(m, c) = (1, 3)$ (14)

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Finally $(m, c) = (1,1), (1,2), (1,3)$



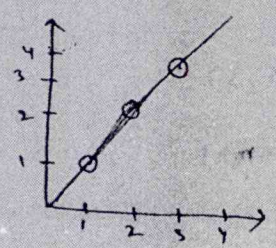
All three lines lie match at ± 1 point so $(m, c) = (1, 0)$

The original equation of line is $y = mx + c$

so put value of ~~m and c~~ $m = 1$ and $c = 0$ so

$\boxed{y = x}$ the ~~the~~ equation of line.

So we are show that point $(1,1), (2,2), (3,3)$ are collinear.



All three point on a single line, so it is colinear.

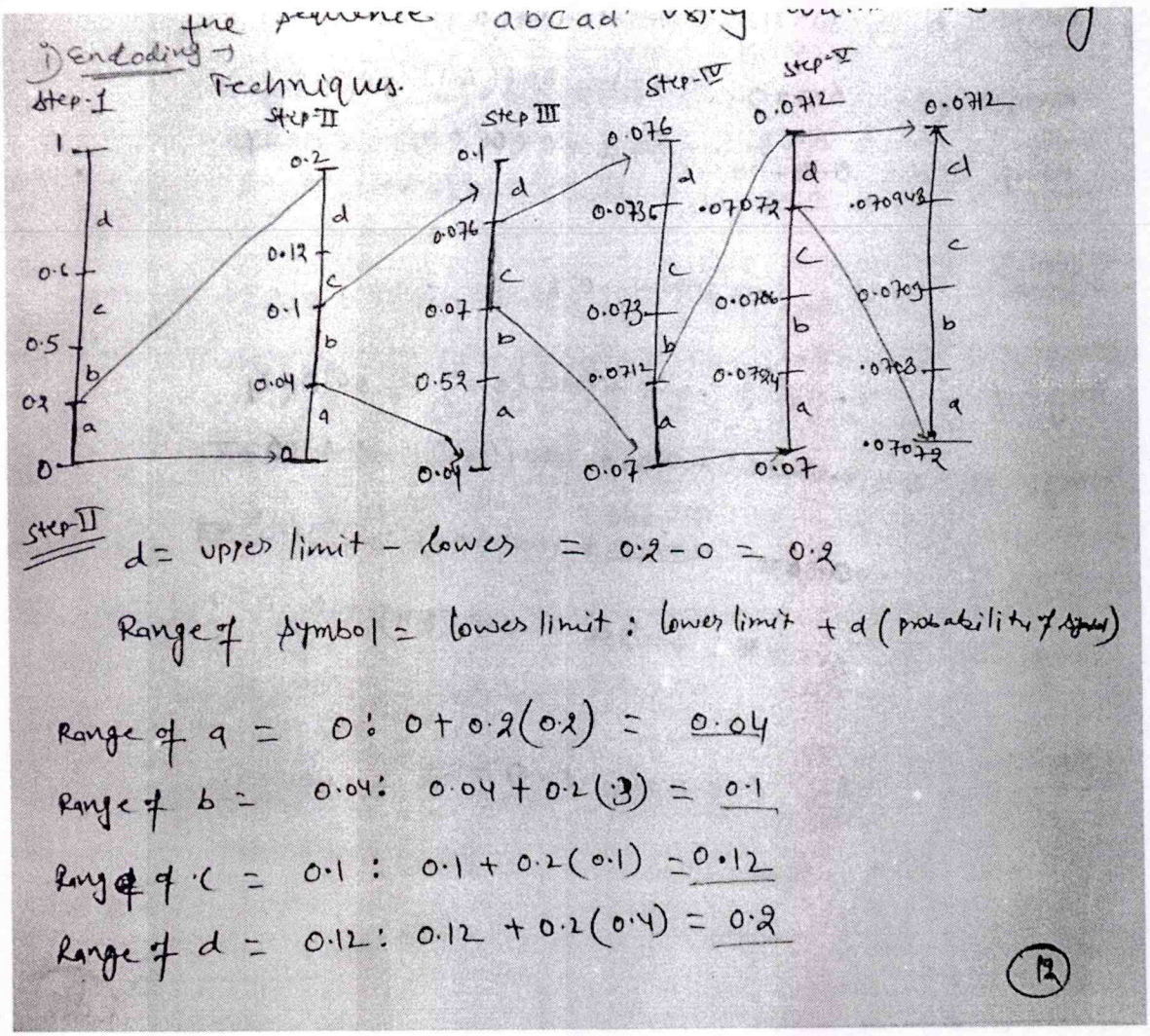
Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Q.6 Consider the probabilities p(a)=0.2, p(b)=0.3, p(c)=0.1, p(d)=0.4. Encode message "abcad", comprising a string by apply arithmetic coding to compress the message?

Solution:



Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Step III →

d = upper limit - lower limit = 0.1 − 0.04 = 0.06

Range of a = 0.04 : 0.04 + 0.06(0.2) = .052
Range of b = 0.052 : .052 + 0.06(0.3) = .07
Range of c = 0.07 : .07 + 0.06 (0.1) = 0.076
Range of d = 0.076 : 0.076 + 0.06 (0.4) = 0.08

Step IV

d = upper limit − lower limit = 0.076 − 0.070 = .006

Range of a = .070 : .07 + .006 (0.2) = 0.0712
Range of b = 0.0712 : 0.0712 + .006 (0.3) = 0.073
Range of c = 0.073 : 0.073 + .006(0.1) = 0.0736
Range of d = 0.0736 : 0.0736 + 0.006(0.4) = 0.076

Step V → d = .0712 − .07 = .0012

Range of a = 0.070 : .070 + .0012 (0.2) = 0.07024
Range of b = 0.07024 : .07024 + .0012 (0.3) = 0.07060
Range of c = 0.0706 : 0.0706 + .0012(0.1) = 0.070732
Range of d = 0.070732 : 0.070732 + .0012(0.4) = 0.0712

Step VI → d = 0.0712 − 0.070732 = .00048

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Range of a = .07072 : .07072 + .00048(0.2) = .0708

Range of b = .0708 : .0708 + .00048(0.3) = .0709

Range of c = .0709 : .0709 + .00048(0.1) = .070948

Range of d = .070948 : .070948 + .00048(0.4) = .0712

Now we will found out tag. and Code word.

→ Here Code Word is

0.070948 < Code word is < 0.0712

$$tag = \frac{\text{upper limit} + \text{lower limit}}{2}$$

$$= \frac{0.0712 + 0.07072}{2}$$

Tag. = 0.07096

Q.7 Determine the Huffman code assignment procedure for the following Data.

Symbol	A	B	C	D	E	F
Probability	0.1	0.4	0.06	0.1	0.04	0.3

Solution: Determine the Huffman code assignment for the given symbols and their probabilities, we can follow these steps:

Step 1: List the Symbols and Their Probabilities

Symbol	A	B	C	D	E	F
Probability	0.1	0.4	0.06	0.1	0.04	0.3

Step 2: Create a Min-Heap of All Symbols

Solution of Question Paper
II Mid-Term Examination, May -2024

Prog./Sem.: B. Tech.-VI	Course Title: DIP	Course Code: 6CAI3-01/6IT3-01
Duration: 1.5 hours	Date: 28.05.2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. S.R. Dogiwal, Mr. Sohan Lal Gupta		

Create nodes for each symbol and their probabilities.
Insert all nodes into a min-heap (priority queue) based on their probabilities.

Step 3: Build the Huffman Tree

First Iteration:

Remove the two nodes with the smallest probabilities (E: 0.04 and C: 0.06).

Combine them to create a new node with probability 0.04+0.06=0.1

Insert the new node back into the min-heap.

Second Iteration:

Remove the two nodes with the smallest probabilities (A: 0.1 and D: 0.1).

Combine them to create a new node with probability 0.1+0.1=0.2.

Insert the new node back into the min-heap.

Third Iteration:

Remove the two nodes with the smallest probabilities (new node 0.1 and F: 0.3).

Combine them to create a new node with probability 0.1+0.3=0.4.

Insert the new node back into the min-heap.

Fourth Iteration:

Remove the two nodes with the smallest probabilities (B: 0.4 and new node 0.2).

Combine them to create a new node with probability 0.4+0.2=0.6.

Insert the new node back into the min-heap.

Fifth Iteration:

Remove the two remaining nodes (new node 0.4 and new node 0.6).

Combine them to create the root node with probability 0.4+0.6=1.0.

Step 4: Assign Huffman Codes:

Traverse the Huffman tree from the root, assigning binary codes to each symbol. Move left for '0' and right for '1'.

Assigning Codes: B: 1 A: 01 D: 00 F: 10 E: 110 C: 111

Summary of Huffman Codes:

Symbol	Probability	Huffman Code
A	0.1	01
B	0.4	1
C	0.06	111
D	0.1	00
E	0.04	110
F	0.3	10

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur		
B.Tech _III__Year _VI__ Semester (Session 2023-2024)		
CO Attainment		
Name of the Exam: Theory Mid term -II	Department: CSE(DS)	Section: DS(A)
Faculty Name: Dr. Arpita Sharma	Course Title with CODE: Cloud Computing (6CDS4-06)	
CO for the course: Upon successful completion of this course, students will be able to:		
CO1: Recall and summarize the basic concepts of cloud computing.		
CO2: Discuss the architectural design of cloud and illustrate various programming models.		
CO3: Outline the virtualization technology and determine their uses.		
CO4: Explain the basic threats and security mechanism in cloud.		
CO5: Summarize the cloud available platforms for business and industry perspective.		

MID TERM EVALUATION

S. NO.	ROLL NO	PART →	A			B		C	Total
		QUESTION NO. →	Q1	Q2	Q3	Q4	Q5	Q6/Q7	
		COURSE OUTCOME(S) SATISFIED →	CO3	CO4	CO5	CO3	CO4	CO5	
		MAXIMUM MARKS →	2	2	2	4	4	6	
		MINIMUM QUALIFYING MARKS (60%) →	1.2	1.2	1.2	2.4	2.4	3.6	
		NAME OF STUDENT ↓							
1	21ESKCX001	Abhi Jain	1	1	2	0	4	3	11
2	21ESKCX002	Abhi Jain	2	2	2	4	4	4	18
3	21ESKCX003	Adamya Agarwal	1	2	2	3	4	4	16
4	21ESKCX004	Aditya Parashar	2	1	2	3	3	4	15
5	21ESKCX005	Aditya Pratap Singh	2	2	2	4	4	2	16
6	21ESKCX006	Aman Sharma	2	2	2	4	4	5	19
7	21ESKCX007	Anugrah Pathak	DB	DB	DB	DB	DB	DB	DB
8	21ESKCX008	Arshia Agarwal	2	1	1	4	2	3	13
9	21ESKCX009	Ayush Soni	2	1	1	4	4	4	16
10	21ESKCX010	Bhagyansh Garg	2	2	2	4	4	4	18
11	21ESKCX011	Bhavya Goyal	2	2	2	4	4	4	18
12	21ESKCX012	Danish Khan	1	1	1	4	4	3	14
13	21ESKCX014	Devansh Pancholi	2	2	2	4	4	4	18
14	21ESKCX015	Dharmanshu Kumar	2	2	2	4	4	5	19
15	21ESKCX016	Dheeraj Saini	2	1	2	4	4	4	17
16	21ESKCX017	Dhruv Pathak	2	2	2	2	4	6	18
17	21ESKCX019	Drishti Gupta	1	2	1	2	3	3	12
18	21ESKCX020	Garv Sharma	2	1	2	4	4	4	17
19	21ESKCX021	Gaurav Sharma	2	1	2	2	4	5	16
20	21ESKCX022	Harsh Dagar	DB	DB	DB	DB	DB	DB	DB
21	21ESKCX023	Harsh Garg	2	1	2	4	4	6	19
22	21ESKCX024	Harsh Goyal	2	2	2	4	4	4	18
23	21ESKCX025	Harsh Tambi	2	2	2	4	3	4	17
24	21ESKCX026	Hridayansh Sharma	1	1	2	2	3	0	9
25	21ESKCX027	Jagrat Parnami	2	2	2	4	4	6	20
26	21ESKCX028	Jatin Yadav	2	2	2	4	4	4	18
27	21ESKCX029	Joshita Sharma	2	2	2	4	3	3	16

28	21ESKCX030	Kanak Soni	2	2	2	3	4	3	**16**
29	21ESKCX031	Karan Choudhary	2	1	1	2	3	2	**11**
30	21ESKCX032	Kartik Sarda	1	1	1	4	4	3	**14**
31	21ESKCX033	Kavya Loona	2	2	2	4	4	4	**18**
32	21ESKCX034	Keshav Sharma	DB	DB	DB	DB	DB	DB	DB
33	21ESKCX035	Khechraay Arkaay	2	2	2	4	4	2	**16**
34	21ESKCX036	Khushi Jain	2	2	2	4	4	4	**18**
35	21ESKCX037	Koopar Vijay	2	2	2	4	4	3	**17**
36	21ESKCX038	Madhav Somani	2	2	2	2	3	4	**15**
37	21ESKCX039	Mehul Sharma	2	2	2	4	4	5	**19**
38	21ESKCX040	Neelesh Jangid	2	2	2	4	4	5	**19**
39	21ESKCX041	Pari Manoj Saraswat	2	1	2	3	4	5	**17**
40	21ESKCX042	Pawan Kumar Jat	2	2	1	3	4	5	**17**
41	21ESKCX043	Prashant Sharma	1	1	2	3	4	2	**13**
42	21ESKCX045	Priyanshu Batra	1	2	1	3	3	4	**14**
43	21ESKCX046	Pulkit Ahuja	2	2	2	4	4	4	**18**
44	21ESKCX047	Rishi Soni	2	2	2	4	4	4	**18**
45	21ESKCX048	Ritin Yadav	DB	DB	DB	DB	DB	DB	DB
46	21ESKCX049	Rohit Soni	2	2	2	4	4	4	**18**
47	21ESKCX050	Sagar Jha	2	1	2	2	4	3	**14**
48	21ESKCX051	Sahil Kawar	2	2	1	3	3	2	**13**
49	21ESKCX052	Simran Udhani	2	2	2	4	4	3	**17**
50	21ESKCX053	Tejasvi	1	1	1	2	4	3	**12**
51	21ESKCX054	Ujjwal Tak	1	1	1	4	3	5	**15**
52	21ESKCX055	Vaishnavi Sharma	2	2	1	4	4	6	**19**
53	21ESKCX056	Vandit Khandelwal	1	1	2	3	3	3	**13**
54	21ESKCX057	Vedeesh Sharma	2	2	2	2	4	2	**14**
55	21ESKCX058	Vishal Bhagwani	DB	DB	DB	DB	DB	DB	DB
56	21ESKCX059	Vishal Sharma	1	2	2	3	4	5	**17**
57	21ESKCX300	Abhijeet Mathur	1	2	2	3	4	5	**17**
58	21ESKCX301	Megha Agarwal	2	2	2	3	3	0	**12**
59	21ESKCX302	Manogya Jain	2	2	2	4	2	5	**17**
60	21ESKCX303	Jashan	2	2	2	4	4	4	**18**
61	21ESKCX304	Rahul Gautam	2	1	1	4	3	3	**14**
62	22ESKCX200	Chahat Faujdar	2	2	2	4	4	6	**20**
63	22ESKCX201	Danish Qamar	2	2	2	3	3	3	**15**
64	22ESKCX202	Irfan Ahmad Dar	2	2	2	3	2	3	**14**
Total Students in Class (S1)			64	64	64	64	64	64	
No. of Students scored >=60% marks (S2)			46	40	46	49	56	36	

CO Attainment Level	**Weighted Avg. Percentage of CO Attainment (N)**	**CO Attainment Level**
Attainment of CO-1	-	-
Attainment of CO-2	-	-

Attainment of CO-3	**75.00%**	**3**
Attainment of CO-4	**79%**	**3**
Attainment of CO-5	**60%**	**2**
Criterion of Percentage for CO Attainment Level considering target 60%	0 < N ≤ 50%	Low (1)
	50% < N ≤ 70%	Medium (2)
	70% < N ≤ 100%	High (3)

Weighted Avg. percentage attainment of CO_j :

$$CO_j = \frac{\sum_{i=1}^{n} S2_i \times MM_i}{\sum_{i=1}^{n} S1_i \times MM_i} \times 100$$

where i = *Index of question number correlated with* CO_j

j = *Index of the course outcome*

n = *Number of questions*

MM = *Maximum marks*

Faculty name with signature: Dr. Arpita Sharma

Sample Calculation

$$CO_1 = \frac{S2_1 \times MM_1 + S2_2 \times MM_2 + S2_4 \times MM_4 + S2_6 \times MM_6}{S1_1 \times MM_1 + S1_2 \times MM_2 + S1_4 \times MM_4 + S1_6 \times MM_6} \times 100$$

$$CO_1 = \frac{44 \times 2 + 33 \times 2 + 6 \times 4 + 20 \times 6}{57 \times 2 + 57 \times 2 + 57 \times 4 + 57 \times 6} \times 100 = 37.34\%$$

$$CO_2 = \frac{S2_3 \times MM_3 + S2_5 \times MM_5}{S1_3 \times MM_3 + S1_5 \times MM_5} \times 100$$

$$CO_2 = \frac{36 \times 2 + 45 \times 4}{57 \times 2 + 57 \times 4} \times 100 = 73.68\%$$

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur		
B.Tech III Year VI Semester (Session 2023-2024)		
CO Attainment		
Name of the Exam: Theory Mid term -II	**Department:CSE**	**Section: A**
Faculty Name: Jeba Nega Cheltha	**Course Title with CODE: Information Security System (6CDS-03)**	
CO for the course: **Upon successful completion of this course, students will be able to:**		
CO1: Understand Security Fundamentals and Encryption Techniques.		
CO2: Apply modern block cipher with modes.		
CO3: Apply Public Key Cryptosystems for Secure Communication.		
CO4: Analyze the fundamental concepts of Cryptographic Hash Functions and Message Authentication Codes.		
CO5: Illustrate Proficiency in Key Management and Web Security.		

MID TERM EVALUATION

S. NO.	ROLL NO	PART →	A			B		C	Total
		QUESTION NO. →	Q1	Q2	Q3	Q4	Q5	Q6/Q7	
		COURSE OUTCOME(S) SATISFIED →	CO3	CO5	CO5	CO3	CO5	CO4	
		MAXIMUM MARKS →	2	2	2	4	4	6	
		MINIMUM QUALIFYING MARKS (60%) →	1.2	1.2	1.2	2.4	2.4	3.6	
		NAME OF STUDENT ↓							
1	21ESKCX001	Abhi Jain	2	2	2	2		3	11
2	21ESKCX002	Abhi Jain	2	2	2	2	3	3	14
3	21ESKCX003	Adamya Agarwal	2	2	2	2	3	5	16
4	21ESKCX004	Aditya Parashar	2	2	2	2	4	6	18
5	21ESKCX005	Aditya Pratap Singh	AB	AB	AB	AB	AB	AB	AB
6	21ESKCX006	Aman Sharma	2	2	2	4	4	6	20
7	21ESKCX007	Anugrah Pathak	DB	DB	DB	DB	DB	DB	DB
8	21ESKCX008	Arshia Agarwal	2	2	2	4	4	5	19
9	21ESKCX009	Ayush Soni	2	2	2	4	4	6	20
10	21ESKCX010	Bhagyansh Garg	2	2	2	4	4	6	20
11	21ESKCX011	Bhavya Goyal	2	2	2	4	3	6	19
12	21ESKCX012	Danish Khan	2	2	2	3	2	3	14
13	21ESKCX014	Devansh Pancholi	2	2	2	4	3	6	19
14	21ESKCX015	Dharmanshu Kumar	2	2	2	4			10
15	21ESKCX016	Dheeraj Saini	2	2	2	4	4	5	19
16	21ESKCX017	Dhruv Pathak	2	2	2	4	4	6	20
17	21ESKCX019	Drishti Gupta	2	1	2	3	4	5	17
18	21ESKCX020	Garv Sharma	2	1	2	4		5	14
19	21ESKCX021	Gaurav Sharma	2	1	2	2		5	12
20	21ESKCX022	Harsh Dagar	DB	DB	DB	DB	DB	DB	DB
21	21ESKCX023	Harsh Garg	2	1	2	2		5	12
22	21ESKCX024	Harsh Goyal	2	1	2	4	3	5	17
23	21ESKCX025	Harsh Tambi	2	1	2		4		9
24	21ESKCX026	Hridayansh Sharma	2	1	2		2		7
25	21ESKCX027	Jagrat Parnami	DB	DB	DB	DB	DB	DB	DB

4/6/24

26	21ESKCX028	Jatin Yadav	1	1	2	4	3	5	**16**
27	21ESKCX029	Joshita Sharma	2	1	2	4	3	6	**18**
28	21ESKCX030	Kanak Soni	2	1	2	4	3	6	**18**
29	21ESKCX031	Karan Choudhary	2	1	2		2	4	**11**
30	21ESKCX032	Kartik Sarda	2	1	2	4	4	3	**16**
31	21ESKCX033	Kavya Loona	2	1	2	4	4	6	**19**
32	21ESKCX034	Keshav Sharma	DB	DB	DB	DB	DB	DB	**DB**
33	21ESKCX035	Khechraay Arkaay	2	1	2	4	3	4	**16**
34	21ESKCX036	Khushi Jain	2	1	2	4	4	2	**15**
35	21ESKCX037	Koopar Vijay	2	1	2	4	2	4	**15**
36	21ESKCX038	Madhav Somani	DB	DB	DB	DB	DB	DB	**DB**
37	21ESKCX039	Mehul Sharma	1	2	2	4	4	5	**18**
38	21ESKCX040	Neelesh Jangid	2	1	2	4	2	3	**14**
39	21ESKCX041	Pari Manoj Saraswat	2	2	2	2	2	4	**14**
40	21ESKCX042	Pawan Kumar Jat	2	i	2	1		2	**8**
41	21ESKCX043	Prashant Sharma	2	1	2	4	2	3	**14**
42	21ESKCX045	Priyanshu Batra	2	1	2	4	4	6	**19**
43	21ESKCX046	Pulkit Ahuja	2	1	2	4	2	5	**16**
44	21ESKCX047	Rishi Soni	2	2	2	4	4	6	**20**
45	21ESKCX048	Ritin Yadav	DB	DB	DB	DB	DB	DB	**DB**
46	21ESKCX049	Rohit Soni	2	2	2	4	3	6	**19**
47	21ESKCX050	Sagar Jha	DB	DB	DB	DB	DB	DB	**DB**
48	21ESKCX051	Sahil Kawar	2	1	2	3		4	**12**
49	21ESKCX052	Simran Udhani	2	2	2	2	2	6	**16**
50	21ESKCX053	Tejasvi	2	1	2	1	3	3	**12**
51	21ESKCX054	Ujjwal Tak	1	1	2	3	1	6	**14**
52	21ESKCX055	Vaishnavi Sharma	2	2	2	4	4	6	**20**
53	21ESKCX056	Vandit Khandelwal	2	1	1	2	0	1	**7**
54	21ESKCX057	Vedeesh Sharma	2	1	2	3	1	2	**11**
55	21ESKCX058	Vishal Bhagwani	DB	DB	DB	DB	DB	DB	**DB**
56	21ESKCX059	Vishal Sharma	1	1	2	4	4	5	**17**
57	21ESKCX300	Abhijeet Mathur	2	1	2	2	2	4	**13**
58	21ESKCX301	Megha Agarwal	1	1	1			3	**6**
59	21ESKCX302	Manogya Jain	2	1	2	4	4	6	**19**
60	21ESKCX303	Jashan	2	1	2	4	3	6	**18**
61	21ESKCX304	Rahul Gautam	2	1	2		3	3	**11**
62	22ESKCX200	Chahat Faujdar	2	1	2	4	4	6	**19**
63	22ESKCX201	Danish Qamar	1	1	1	2	3	2	**10**
64	22ESKCX202	Irfan Ahmad Dar	2	1	2	3			**8**
Total Students in Class (S1)			64	64	64	64	64	64	
No. of Students scored >=60% marks (S2)			49	20	52	37	33	37	

4/6/24

CO Attainment Level	Weighted Avg. Percentage of CO Attainment (N)	CO Attainment Level
Attainment of CO-1	-	-
Attainment of CO-2	-	-
Attainment of CO-3	64.06%	2
Attainment of CO-4	57.81%	2
Attainment of CO-5	53.91%	2
Criterion of Percentage for CO Attainment Level considering target 60%	0 < N ≤ 50%	Low (1)
	50% < N ≤ 70%	Medium (2)
	70% < N ≤ 100%	High (3)

Weighted Avg. percentage attainment of CO_j :

$$CO_j = \frac{\sum_{i=1}^{n} S2_i \times MM_i}{\sum_{i=1}^{n} S1_i \times MM_i} \times 100$$

where i = *Index of question number correlated with* CO_j

j = *Index of the course outcome*

n = *Number of questions*

MM = *Maximum marks*

JEBA NEGA CHELLA 4/6/23

Faculty name with signature

Sample Calculation

$$CO_1 = \frac{S2_1 \times MM_1 + S2_2 \times MM_2 + S2_4 \times MM_4 + S2_6 \times MM_6}{S1_1 \times MM_1 + S1_2 \times MM_2 + S1_4 \times MM_4 + S1_6 \times MM_6} \times 100$$

$$CO_1 = \frac{44 \times 2 + 33 \times 2 + 6 \times 4 + 20 \times 6}{57 \times 2 + 57 \times 2 + 57 \times 4 + 57 \times 6} \times 100 = 37.34\%$$

$$CO_2 = \frac{S2_3 \times MM_3 + S2_5 \times MM_5}{S1_3 \times MM_3 + S1_5 \times MM_5} \times 100$$

$$CO_2 = \frac{36 \times 2 + 45 \times 4}{57 \times 2 + 57 \times 4} \times 100 = 73.68\%$$

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

II -Mid Term Examination, June-2024

Prog./Semester:	M. Tech.-II	Branch:	CSE
Course Title:	DPA	Course Code:	2MCS1-02
Duration:	1.5 Hours	Maximum Marks:	20
Session : Second		Roll No.:	
Instructions if any:			

PART -A (short answer type questions)
(All questions are compulsory)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
1	Illustrate the lockout-free algorithm in DS	2	3	2
2	Explain the transformations from shared memory model to network model in DS.	2	4	2
3	Outline the few of real-world applications that benefit significantly from parallel processing?	2	5	2

PART -B (Analytical/Problem solving questions)
(All questions are compulsory)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
4	Classify common bottlenecks in parallel divide-and-conquer algorithms, and how can they be mitigated?	4	5	4
5	Compare the Bakery BurnsME algorithm with other mutual exclusion algorithms in terms of efficiency and complexity?	4	3	4

PART- C (Descriptive/Analytical/Problem solving/Design questions)
(Attempt any one Question)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
6	Demonstrate the key challenges in designing a PRAM-based graph coloring algorithm	6	6	3
7	Demonstrate prefix sum problem, and how can it be efficiently solved using parallel algorithms on the PRAM model?	6	6	3

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, June -2024

Prog./Sem.: M. Tech. II Sem.	Course Title: DPA	Course Code:2MCS1-02
Duration: 1.5 hours	Date: 10/6/2024 Session : SECOND	Max Marks: 20
Submitted By: Dr. Nilam		

S. No.	Course Outcome	Percentage Coverage
CO1	**Identify** the concepts of parallel processing and understand the particular problems arising in the programming of parallel machines	-
CO2	**Compare and Evaluate** the parallel computing models and the "parallel-way of thinking" required in the design of parallel algorithms	-
CO3	**Apply** the basic algorithmic techniques and design algorithms in shared memory as well as distributed memory environment	30
CO4	**Acquire** knowledge of those problems which have been perceived as intractable for parallelization	10
CO5	**Understand** and be able to apply basic parallel programming principles in a shared/distributed memory environment	30
CO6	**Classify** the strengths and limitations of parallel computing approaches to problem-solving	30

S. No.	Bloom's Level	Percentage Coverage
1	BL1- Remember	-
2	BL2- Understand	28.57%
3	BL3- Apply	28.57%
4	BL4- Analyze	42.85%
5	BL5- Evaluate	-
6	BL6- Create	-

-- Question Paper Solution --

Q.1. Illustrate the lockout-free algorithm in DS

Ans. The lockout-free algorithm in data structures represents a significant advancement in concurrent programming, particularly in scenarios where multiple processes contend for shared resources. Unlike traditional locking mechanisms, which can lead to situations where a process is indefinitely blocked or

"locked out" by others, lockout-free algorithms ensure that every process eventually makes progress.

At its core, a lockout-free algorithm employs strategies such as optimistic concurrency control and non-blocking synchronization techniques to enable processes to proceed independently whenever possible. This is achieved by allowing processes to perform their operations without waiting for exclusive access to a resource. If contention occurs, the algorithm dynamically resolves conflicts by detecting and mitigating interference among processes.

One common implementation of lockout-free algorithms is through techniques like compare-and-swap (CAS) operations, which enable atomic updates to shared variables without the need for traditional locks. By leveraging CAS and similar primitives, lockout-free algorithms ensure progress even in the presence of contention, ultimately leading to improved scalability and efficiency in concurrent systems.

Q.2. Explain the transformations from shared memory model to network model in DS.

Ans. The transformations from a shared memory model to a network model in data structures involve adapting the communication and synchronization mechanisms to accommodate distributed environments. In a shared memory model, processes interact by accessing a common address space, simplifying communication and synchronization. However, in a network model, processes reside on separate machines connected by a network, requiring a different approach to achieve coordination and data sharing.

To transition from a shared memory model to a network model, several transformations are necessary. Firstly, communication primitives such as message passing protocols replace shared memory operations to enable inter-process communication over the network. Processes must exchange messages to exchange data and synchronize their actions. Additionally, synchronization mechanisms like locks and barriers need to be reengineered to work across distributed systems. For instance, distributed locking protocols ensure mutual exclusion by coordinating access to shared resources across multiple machines.

Moreover, fault tolerance becomes a crucial consideration in the network model due to the potential for network failures and process crashes. Techniques like replication and consensus algorithms are employed to ensure data consistency and system resilience in the face of failures. Furthermore, distributed algorithms often prioritize scalability and efficiency to accommodate the distributed nature of the system, leading to the development of decentralized and parallelized approaches.

Q.3. **Outline the few of real-world applications that benefit significantly from parallel processing?**

Ans. Several real-world applications benefit significantly from parallel processing, leveraging its ability to

divide tasks among multiple processors for improved performance and efficiency. High-performance computing (HPC) applications such as weather forecasting, climate modeling, and computational fluid dynamics rely on parallel processing to analyze vast amounts of data and perform complex simulations in a timely manner. In scientific research, parallel processing accelerates tasks like genomic sequencing, protein folding simulations, and drug discovery by distributing computations across multiple cores or nodes. Additionally, parallel processing enhances multimedia applications such as video encoding, image processing, and real-time rendering in gaming and animation industries, enabling faster production times and higher-quality outputs. Data analytics and machine learning tasks, including big data processing, neural network training, and natural language processing, benefit from parallel processing to handle large datasets and complex algorithms efficiently. Overall, parallel processing revolutionizes various domains by unlocking computational power and enabling faster, more scalable, and resource-efficient solutions to complex problems.

Q.4. **Classify common bottlenecks in parallel divide-and-conquer algorithms, and how can they be mitigated?**

Ans. Common bottlenecks in parallel divide-and-conquer algorithms can arise due to various factors, including communication overhead, load imbalance, and synchronization issues:

1. Communication Overhead: In parallel divide-and-conquer algorithms, excessive communication between processors can lead to performance bottlenecks, especially when sharing intermediate results or coordinating tasks. This overhead can slow down the overall execution time and reduce scalability.

2. Load Imbalance: Uneven distribution of work among processors can result in load imbalance, where some processors complete their tasks much faster than others, leading to idle time and inefficient resource utilization.

3. Synchronization: Synchronization points, such as barrier synchronization or collective operations, can introduce delays as processors wait for others to complete their tasks before proceeding. Poorly designed synchronization mechanisms can lead to contention and performance degradation.

To mitigate these bottlenecks, several strategies can be employed:

1. Minimize Communication: Reduce unnecessary communication by optimizing data sharing and minimizing the exchange of intermediate results between processors. This can be achieved by carefully designing data partitioning schemes and limiting the scope of shared data.

2. Load Balancing: Implement dynamic load balancing techniques to evenly distribute workloads among processors and

minimize idle time. This may involve dynamically reallocating tasks based on workload characteristics or adjusting task granularity to ensure balanced computation.

3. Asynchronous Execution: Utilize asynchronous execution whenever possible to overlap computation with communication and reduce synchronization overhead. Asynchronous algorithms allow processors to work independently without waiting for others, improving overall throughput and reducing latency.

4. Optimized Synchronization: Optimize synchronization mechanisms to minimize contention and overhead. This can involve using lightweight synchronization primitives, such as non-blocking synchronization or fine-grained locking, and carefully designing synchronization points to avoid unnecessary delays.

5. Scalable Data Structures: Employ scalable data structures and algorithms that can efficiently handle concurrent access and scale to large problem sizes. This may involve using data structures designed for parallel environments, such as parallel data structures or lock-free data structures, to minimize contention and maximize throughput.

By addressing these bottlenecks through careful algorithm design, optimization, and parallelization techniques, the performance of parallel divide-and-conquer algorithms can be significantly improved, leading to better scalability and efficiency in parallel computing environments.

Q.5. Compare the Bakery BurnsME algorithm with other mutual exclusion algorithms in terms of efficiency and complexity?

Ans. The Bakery and BurnsME algorithms are both mutual exclusion algorithms designed to ensure that only one process accesses a shared resource at a time, preventing data corruption and race conditions in concurrent systems. However, they differ in their approaches to achieving mutual exclusion, which can affect their efficiency and complexity:

1. Bakery Algorithm:

- Efficiency: The Bakery algorithm is relatively simple and easy to understand, making it suitable for small-scale systems. However, its performance may degrade in large-scale systems due to its O(n) complexity for n processes, where each process must obtain a unique ticket before entering the critical section.

- Complexity: The Bakery algorithm guarantees fairness by assigning unique tickets to processes based on their arrival order. While this ensures that every process eventually enters the critical section, it introduces additional complexity and overhead for managing ticket acquisition and comparison, especially in highly concurrent environments.

2. BurnsME Algorithm:

- Efficiency: The BurnsME algorithm is designed to improve efficiency by reducing the overhead associated with mutual exclusion. By utilizing a combination of optimistic concurrency control and adaptive contention management, BurnsME aims

to minimize unnecessary delays and contention, leading to improved overall performance.

- Complexity: BurnsME achieves efficiency by dynamically adapting its behavior based on contention levels and workload characteristics. While this approach reduces overhead and improves scalability, it also introduces additional complexity in managing contention and resolving conflicts, potentially making the algorithm harder to implement and analyze compared to simpler approaches like Bakery.

Q.6. Demonstrate the key challenges in designing a PRAM-based graph coloring algorithm.

Ans. Designing a PRAM-based graph coloring algorithm presents several key challenges due to the distributed nature of the PRAM model and the complexities of graph structures:

1. Data Distribution: In a PRAM-based algorithm, the graph data must be distributed across multiple processors. Determining an efficient and balanced distribution strategy is challenging, especially for large graphs with irregular structures. Uneven distribution can lead to load imbalance and inefficiencies in parallel execution.

2. Concurrency Control: Coordinating access to shared data, such as the graph structure and color assignments, among multiple processors presents a significant challenge. Traditional locking mechanisms may introduce contention and synchronization overhead, impacting performance. Designing efficient and scalable concurrency control mechanisms that minimize contention while ensuring correctness is crucial.

3. Communication Overhead: Graph coloring algorithms often require processors to exchange information about neighboring vertices to enforce color constraints and avoid conflicts. However, excessive communication can lead to high communication overhead, especially in dense or highly connected graphs. Minimizing communication while maintaining global consistency is a non-trivial task.

4. Scalability: Ensuring the scalability of a PRAM-based graph coloring algorithm is essential for handling increasingly large and complex graphs. As the number of processors increases, the algorithm should scale efficiently to leverage additional computational resources without sacrificing performance or introducing bottlenecks.

5. Conflict Resolution: Resolving conflicts between neighboring vertices competing for the same color is a fundamental aspect of graph coloring algorithms. In a PRAM-based approach, efficiently identifying and resolving conflicts in a distributed and parallel manner poses a significant challenge. Ensuring that conflict resolution mechanisms are both effective and scalable is crucial for algorithm performance.

6. Fault Tolerance: Dealing with processor failures or network partitions in a distributed PRAM system adds another layer of complexity to algorithm design. Implementing fault-tolerant mechanisms to handle failures gracefully and maintain algorithm correctness and progress is essential for robustness in real-world distributed environments.

Solution of Question Paper
II Mid-Term Examination, June -2024

Prog./Sem.: M. Tech. II Sem.	**Course Title: DPA**	**Course Code:2MCS1-02**
Duration: 1.5 hours	**Date: 10/6/2024** **Session : SECOND**	**Max Marks: 20**
Submitted By: Dr. Nilam		

Addressing these challenges requires careful algorithm design, leveraging parallelism, optimization techniques, and distributed computing principles to develop efficient and scalable PRAM-based graph coloring algorithms capable of handling large-scale and complex graph structures effectively.

Q.7. Demonstrate prefix sum problem, and how can it be efficiently solved using parallel algorithms on the PRAM model?

Ans. The prefix sum problem involves computing the cumulative sum of elements in an array, where each element of the output array is the sum of all preceding elements in the input array. This problem is commonly encountered in various parallel computing applications, such as parallel reduction and parallel scan operations.

In the PRAM model, parallel algorithms can efficiently solve the prefix sum problem by leveraging concurrent processing and shared memory access among multiple processors. One of the widely used parallel algorithms for prefix sum is the parallel scan algorithm.

The parallel scan algorithm can be efficiently implemented using the following steps in the PRAM model:

1. Partitioning: Divide the input array into smaller chunks, distributing them evenly across the available processors. Each processor is responsible for computing the prefix sum of its assigned chunk.

2. Local Prefix Sum: Each processor computes the prefix sum of its local chunk using a sequential prefix sum algorithm, such as the iterative or recursive approach. This step generates the local prefix sum array for each chunk.

3. Parallel Reduction: Perform a parallel reduction operation to compute the sum of the last element of each local prefix sum array. This operation involves recursively halving the number of processors and adding corresponding elements pairwise until only one processor remains with the final result.

4. Global Offset Calculation: Broadcast the final sum obtained from the parallel reduction to all processors. Each processor then calculates its global offset by summing the local prefix sum of the previous chunks and adding the received final sum.

5. Update Prefix Sums: Each processor updates its local prefix sum array by adding its global offset to each element. This step ensures that the prefix sum of each element is correctly computed with respect to the entire input array.

By following these steps, the parallel scan algorithm efficiently computes the prefix sum of the input array in parallel on the PRAM model. This approach minimizes communication overhead and maximizes parallelism, leading to significant improvements in performance and scalability compared to sequential algorithms, especially for large input sizes.

-----End-----

Swami Keshvanand Institute of Technology, Management &Gramothan, Jaipur

II-Mid Term Examination, June-2024

Prog./Semester:	M. Tech.-II	Branch:	CSE
Course Title:	Data Science	Course Code:	2MCS1-01
Duration:	1.5 Hours	Maximum Marks:	20
Session : First		Roll No.:	
Instructions if any:			

PART -A (short answer type questions)

(All questions are compulsory)

Q. No.	Question	Max. Marks	CO(s)	Bloom's Level
1	Define the challenges associated with collecting data from IoT devices? How can these challenges be addressed?	2	CO5	L2
2	Discuss the importance of handling missing data in datasets. What are some techniques used to handle missing values?	2	CO5	L2
3	List the common data pre-processing steps required before performing data analysis? Explain each step briefly.	2	CO5	L2

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Solution of Question Paper
II Mid-Term Examination, June -2024

Prog./Sem.: M.tech /II	**Course Title: Data Science**	**Course Code:2MCS1-01**
Duration: 1.5 hours	**Date: 10,June,2024** **Session : FIRST**	**Max Marks: 20**
Submitted By: Dr. Meenakshi Nawal		

S. No.	Course Outcome	Percentage Coverage
CO1	Building the fundamentals of data science.	N.A
CO2	Imparting design thinking capability to build big-data.	N.A
CO3	Developing design skills of models for big data problems .	20
CO4	Gaining practical experience in programming tools for data sciences.	20
CO5	Empowering students with tools and techniques used in data science.	60

S. No.	Bloom's Level	Percentage Coverage
1	BL1- Remember	-
2	BL2- Understand	30%
3	BL3- Apply	50%
4	BL4- Analyze	20%
5	BL5- Evaluate	-
6	BL6- Create	-

-- Question Paper Solution --

Solution 1: Collecting data from IoT (Internet of Things) devices presents several challenges due to the unique characteristics and requirements of these devices. Here are some key challenges and potential solutions:

Challenges

1. **Data Volume and Velocity**
   - **Challenge:** IoT devices generate massive amounts of data at high speed, leading to challenges in real-time data processing, storage, and analysis.
   - **Solution:** Implement scalable cloud storage solutions and use edge computing to process data closer to the source. Utilize streaming data platforms like Apache Kafka for real-time data handling.
2. **Data Variety and Integration**
   - **Challenge:** IoT devices produce diverse types of data (e.g., structured, semi-structured, unstructured) from various sources, making integration complex.
   - **Solution:** Use data integration platforms and middleware to standardize and consolidate data from different sources. Employ data lakes to store diverse data types and facilitate integration.
3. **Data Quality and Reliability**
   - **Challenge:** IoT data can be noisy, incomplete, or inconsistent due to sensor errors, network issues, or device malfunctions.
   - **Solution:** Implement robust data cleaning and preprocessing techniques. Use redundancy and

Prog./Sem.: M.tech /II	Course Title: Data Science	Course Code:2MCS1-01
Duration: 1.5 hours	Date: 10,June,2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. Meenakshi Nawal		

error-checking mechanisms to ensure data accuracy and reliability.

4. **Data Security and Privacy**
   - **Challenge:** IoT devices are often vulnerable to cyber-attacks, and the data they collect can be sensitive, raising privacy concerns.
   - **Solution:** Use strong encryption methods for data transmission and storage. Implement secure authentication and authorization protocols. Adhere to data privacy regulations (e.g., GDPR) and ensure anonymization where necessary.
5. **Scalability**
   - **Challenge:** The number of IoT devices can grow rapidly, necessitating scalable infrastructure for data collection and processing.
   - **Solution:** Adopt cloud-based solutions that offer elastic scalability. Use microservices architecture to handle the growing number of devices and data streams.
6. **Latency and Real-Time Processing**
   - **Challenge:** Some applications require real-time data processing, and any latency can impact performance and decision-making.
   - **Solution:** Implement edge computing to process data locally and reduce latency. Use real-time data processing frameworks such as Apache Flink or Apache Storm.
7. **Interoperability**
   - **Challenge:** IoT devices from different manufacturers may use different communication protocols, leading to interoperability issues.
   - **Solution:** Use standardized protocols (e.g., MQTT, CoAP) and adopt IoT platforms that support multiple protocols. Develop APIs and use middleware to facilitate communication between heterogeneous devices.
8. **Power Consumption**
   - **Challenge:** IoT devices, especially those that are battery-powered, need to be energy-efficient to prolong their operational life.
   - **Solution:** Optimize data transmission intervals and use low-power communication protocols. Implement energy-efficient hardware and software design practices.

Addressing the Challenges

1. **Scalable Architecture**
   - **Implement cloud and edge computing solutions to handle large volumes of data and provide real-time processing capabilities.**
   - **Use distributed databases and data storage solutions that can scale horizontally to accommodate growing data needs.**
2. **Data Integration and Standardization**
   - **Adopt data integration platforms and middleware to unify data formats and ensure seamless integration from various IoT devices.**
   - **Implement data lakes to store raw data from multiple sources, allowing for flexible and comprehensive data analysis.**
3. **Data Quality Management**
   - **Develop robust data validation and cleaning processes to ensure high-quality data.**
   - **Use machine learning algorithms to detect and correct anomalies in real-time.**

Solution of Question Paper
II Mid-Term Examination, June -2024

Prog./Sem.: M.tech /II	Course Title: Data Science	Course Code:2MCS1-01
Duration: 1.5 hours	Date: 10,June,2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. Meenakshi Nawal		

4. **Security and Privacy Measures**
   - **Encrypt data at rest and in transit to protect against unauthorized access.**
   - **Implement strong access control mechanisms and regularly update security protocols to address emerging threats.**
   - **Adopt privacy-by-design principles to ensure compliance with data protection regulations.**
5. **Standardization and Interoperability**
   - **Promote the use of standardized communication protocols and interoperable IoT platforms.**
   - **Develop APIs and use middleware solutions to facilitate communication between devices from different manufacturers.**
6. **Energy Efficiency**
   - **Optimize data transmission and processing schedules to reduce power consumption.**
   - **Utilize energy-efficient hardware and implement software optimizations to extend battery life.**

By addressing these challenges with appropriate strategies and technologies, the effective collection, processing, and analysis of data from IoT devices can be achieved, enabling the full potential of IoT applications across various domains.

Solution 2: Handling missing data in datasets is crucial for ensuring the quality and reliability of data analysis and model building. Missing data can lead to biased estimates, reduced statistical power, and invalid conclusions. Properly addressing missing data helps maintain the integrity of the dataset and improves the accuracy of predictive models. Here are some key reasons why handling missing data is important and several techniques used to handle missing values:

Importance of Handling Missing Data

1. **Accuracy of Analysis**
   - Missing data can introduce bias, leading to inaccurate statistical estimates and misleading conclusions. Proper handling ensures that the analysis reflects the true underlying patterns in the data.
2. **Model Performance**
   - Machine learning models trained on datasets with missing values can perform poorly, as the missing data can distort the learned relationships between variables. Handling missing values improves model accuracy and generalization.
3. **Data Integrity**
   - Ensuring that datasets are complete and consistent is essential for maintaining data integrity. Handling missing values helps in creating a clean and reliable dataset for analysis.
4. **Statistical Power**
   - Missing data can reduce the statistical power of an analysis, making it harder to detect

significant effects. Proper handling of missing data helps retain the maximum amount of information from the dataset.

5. **Comparability**
   - For longitudinal studies or time series data, handling missing values is crucial to maintain comparability across different time points or samples.

Techniques to Handle Missing Values

1. **Deletion Methods**
   - **Listwise Deletion:** Remove any records (rows) that have one or more missing values. This method is simple but can lead to a significant loss of data, especially if the missing values are frequent.
   - **Pairwise Deletion:** Use all available data without discarding entire records. Each analysis or computation is performed using the available data for the variables involved. This can be useful but may lead to inconsistent sample sizes.
2. **Imputation Methods**
   - **Mean/Median/Mode Imputation:** Replace missing values with the mean, median, or mode of the available data. This method is straightforward but can reduce the variance of the data.
   - **Regression Imputation:** Use regression models to predict and replace missing values based on other available variables. This method considers relationships between variables but can introduce bias if the model is not accurate.
   - **K-Nearest Neighbors (KNN) Imputation:** Replace missing values based on the values of the nearest neighbors. This method is more flexible and can capture local data patterns but may be computationally intensive.
   - **Multiple Imputation:** Create multiple complete datasets by imputing missing values several times, analyze each dataset separately, and then combine the results. This method provides more accurate and unbiased estimates by considering the uncertainty of the imputed values.
3. **Advanced Methods**
   - **Expectation-Maximization (EM):** An iterative method that estimates the missing values by maximizing the likelihood function. It is effective for handling complex data structures but can be computationally intensive.
   - **Machine Learning Models:** Use models such as decision trees or random forests to predict and impute missing values. These models can capture complex relationships between variables and provide accurate imputations.
4. **Data Augmentation**
   - **Synthetic Data Generation:** Create synthetic data points to replace missing values based on the statistical properties of the existing data. This method helps retain data patterns and variability.

Practical Considerations

- **Assess the Missing Data Mechanism:** Understand whether the missing data is Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR). The choice of

technique may vary depending on the underlying mechanism.

- **Impact on Analysis:** Consider the impact of the chosen method on the analysis. Some methods may introduce bias or distort the original data distribution.
- **Domain Knowledge:** Leverage domain knowledge to make informed decisions about handling missing data. Domain expertise can provide insights into the likely reasons for missing data and appropriate imputation methods.
- **Software Tools:** Utilize software tools and libraries that offer advanced imputation techniques, such as scikit-learn in Python or the mice package in R.

By carefully selecting and applying appropriate techniques to handle missing values, data scientists can ensure the quality and reliability of their datasets, leading to more accurate and robust analyses and models.

Solution 3: Data preprocessing is a crucial step in the data analysis pipeline. It involves preparing and transforming raw data into a format suitable for analysis. Here are the common data preprocessing steps required before performing data analysis:

1. Data Cleaning

Data cleaning involves identifying and correcting errors and inconsistencies in the data to improve its quality. This step includes:

- **Handling Missing Values:** Missing values can be addressed through deletion, imputation (e.g., using mean, median, mode, or more advanced techniques like KNN or multiple imputation), or flagging.
- **Removing Duplicates:** Duplicate records are identified and removed to avoid bias in the analysis.
- **Correcting Errors:** Errors in data entry (such as typos) are corrected. This may involve checking for out-of-range values or inconsistencies in categorical data.
- **Standardizing Formats:** Ensuring consistency in data formats (e.g., date formats, text case) across the dataset.

2. Data Integration

Data integration combines data from different sources to create a unified dataset. This step includes:

- **Merging Datasets:** Combining multiple datasets into a single dataset based on common keys or attributes.
- **Resolving Conflicts:** Addressing conflicts and discrepancies between data from different sources, such as different naming conventions or measurement units.

3. Data Transformation

Data transformation involves converting data into a suitable format or structure for analysis. This step includes:

- **Normalization and Standardization:** Scaling numerical data to a common range or distribution. Normalization scales data to a range (e.g., 0 to 1), while standardization scales data to have a mean of 0 and a standard deviation of 1.
- **Encoding Categorical Variables:** Converting categorical data into numerical format using techniques such as one-hot encoding, label encoding, or binary encoding.
- **Log Transformation:** Applying logarithmic transformation to skewed data to normalize the distribution and reduce the impact of outliers.

4. Feature Engineering

Feature engineering involves creating new features or modifying existing ones to improve the performance of machine learning models. This step includes:

- **Feature Creation:** Creating new features from existing data, such as combining date and time features into a single datetime feature or extracting textual features.
- **Feature Selection:** Identifying and selecting the most relevant features for the analysis, reducing dimensionality and improving model performance. Techniques include correlation analysis, mutual information, and using algorithms like LASSO.
- **Feature Extraction:** Reducing the number of features while retaining important information, using techniques such as Principal Component Analysis (PCA) or t-SNE.

5. Data Reduction

Data reduction aims to simplify the dataset by reducing its size without losing significant information. This step includes:

- **Sampling:** Selecting a representative subset of the data for analysis, which can be useful when working with very large datasets.
- **Aggregation:** Summarizing data by aggregating it into higher-level units, such as calculating the mean or sum of groups of data.
- **Dimensionality Reduction:** Techniques like PCA or Singular Value Decomposition (SVD) to reduce the number of features while retaining most of the variance in the data.

6. Data Discretization

Data discretization converts continuous data into discrete bins or categories, which can be useful for certain types of analysis. This step includes:

- **Binning:** Dividing continuous variables into intervals or bins, which can be equal-width (each bin has the same range) or equal-frequency (each bin has the same number of observations).
- **Quantization:** Converting continuous values into discrete values based on certain thresholds or criteria.

Prog./Sem.: M.tech /II	Course Title: Data Science	Course Code:2MCS1-01
Duration: 1.5 hours	Date: 10,June,2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. Meenakshi Nawal		

7. Handling Imbalanced Data

In classification problems, handling imbalanced data involves ensuring that the classes are adequately represented. This step includes:

- **Resampling Techniques:** Oversampling the minority class (e.g., using SMOTE) or undersampling the majority class to balance the class distribution.
- **Algorithmic Techniques:** Using algorithms that can handle imbalanced data, such as ensemble methods or cost-sensitive learning.

8. Data Splitting

Data splitting involves dividing the dataset into training and testing sets (and sometimes a validation set) to evaluate the performance of machine learning models. This step includes:

- **Train-Test Split:** Typically splitting the data into 70-80% for training and 20-30% for testing.
- **Cross-Validation:** Dividing the data into k subsets and using each subset as a test set while training on the remaining subsets, repeating this process k times.

Summary

Data preprocessing is an essential step that transforms raw data into a clean and suitable format for analysis. Each of these steps—data cleaning, integration, transformation, feature engineering, data reduction, discretization, handling imbalanced data, and data splitting—plays a vital role in ensuring the quality and effectiveness of the subsequent data analysis and modeling efforts. Properly preprocessed data leads to more accurate and reliable analytical results and machine learning models.

Solution 4:

Creating a choropleth map to visualize demographic data involves several key steps, including data preparation, preprocessing, and visualization. Here's a step-by-step guide to preparing the data and creating a choropleth map:

Step 1: Collect Data

- **Demographic Data:** Obtain demographic data such as population density, income levels, etc., from reliable sources (e.g., government databases, statistical agencies, open data portals).
- **Geospatial Data:** Acquire geographic boundaries for the regions of interest (e.g., countries, states, counties). This can be in the form of shapefiles, GeoJSON files, or other geospatial data formats.

Step 2: Data Preparation

1. **Load the Data:**

- Load demographic data and geospatial data into your analysis environment (e.g., Python, R).
- Example in Python:

```
python
Copy code
import pandas as pd
import geopandas as gpd

demographic_data = pd.read_csv('demographic_data.csv')
geospatial_data = gpd.read_file('geospatial_data.shp')
```

2. **Merge Data:**
   - Ensure both datasets have a common key to merge on (e.g., region code, name).
   - Merge the demographic data with the geospatial data.
   - Example:

```
python
Copy code
merged_data = geospatial_data.merge(demographic_data, on='region_code')
```

3. **Handle Missing Values:**
   - Check for and handle any missing values in the demographic data.
   - Example:

```
python
Copy code
merged_data = merged_data.dropna(subset=['population_density',
'income_level'])
```

Step 3: Data Preprocessing

1. **Normalize Data:**
   - Normalize demographic data if necessary, especially if there are large variations.
   - Example (optional):

```
python
Copy code
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
merged_data[['population_density', 'income_level']] =
scaler.fit_transform(merged_data[['population_density', 'income_level']])
```

2. **Prepare Data for Visualization:**
   - Ensure the merged data is in a format suitable for creating a choropleth map.

Prog./Sem.: M.tech /II	Course Title: Data Science	Course Code:2MCS1-01
Duration: 1.5 hours	Date: 10,June,2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. Meenakshi Nawal		

Step 4: Create the Choropleth Map

1. **Choose Visualization Library:**
   - Select a visualization library that supports choropleth maps (e.g., Folium, Plotly, Matplotlib, Geopandas).
2. **Create the Map:**
   - Use the chosen library to create the choropleth map.
   - Example using Folium:

```
python
Copy code
import folium

# Initialize the map centered around the region of interest
m = folium.Map(location=[latitude, longitude], zoom_start=6)

# Create the choropleth map
folium.Choropleth(
    geo_data=merged_data,
    name='choropleth',
    data=merged_data,
    columns=['region_code', 'population_density'],
    key_on='feature.properties.region_code',
    fill_color='YlGnBu',
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Population Density'
).add_to(m)

# Add a layer control panel
folium.LayerControl().add_to(m)

# Save the map as an HTML file
m.save('choropleth_map.html')
```

3. **Customize the Map:**
   - Customize the map with additional features such as tooltips, color scales, and legends.
   - Example of adding tooltips:

```
python
Copy code
folium.Choropleth(
    geo_data=merged_data,
    name='choropleth',
    data=merged_data,
    columns=['region_code', 'population_density'],
    key_on='feature.properties.region_code',
    fill_color='YlGnBu',
    fill_opacity=0.7,
    line_opacity=0.2,
```

```
    legend_name='Population Density'
).add_to(m)

# Add tooltips
folium.GeoJson(
    merged_data,
    style_function=lambda x: {'fillColor': 'transparent'},
    tooltip=folium.GeoJsonTooltip(fields=['region_name',
'population_density'])
).add_to(m)
```

Step 5: Validate and Save the Map

1. **Validate the Map:**
   - Review the map to ensure it accurately represents the data and the regions are correctly colored according to their demographic values.
2. **Save the Map:**
   - Save the map in a suitable format (e.g., HTML, PNG).
   - Example:

```
python
Copy code
m.save('choropleth_map.html')
```

Summary

Creating a choropleth map involves collecting and preparing both demographic and geospatial data, merging the datasets, preprocessing the data (handling missing values and normalization), and using a visualization library to create and customize the map. The final map can be saved and shared for analysis and presentation. This process helps visualize geographic distributions and patterns in demographic data, aiding in better understanding and decision-making.

Solution 5 : Interpreting machine learning models through visualization is crucial for understanding the model's behavior, validating its predictions, and ensuring transparency. Here are some common techniques used for model interpretation, along with how they can be visualized:

1. Feature Importance Plots

Explanation:

Feature importance plots help identify which features have the most influence on the model's predictions. This is particularly useful for models like decision trees, random forests, and gradient boosting machines.

Prog./Sem.: M.tech /II	Course Title: Data Science	Course Code:2MCS1-01
Duration: 1.5 hours	Date: 10,June,2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. Meenakshi Nawal		

Visualization:

- **Bar Plots:** A common way to visualize feature importance is through bar plots, where each bar represents the importance score of a feature.
- **Example:**

```
python
Copy code
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming feature_importances_ is an array of importance scores
feature_importances = model.feature_importances_
features = ['feature1', 'feature2', 'feature3']  # replace with actual feature
names

importance_df = pd.DataFrame({'Features': features, 'Importance':
feature_importances})
importance_df = importance_df.sort_values(by='Importance', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Features', data=importance_df)
plt.title('Feature Importance')
plt.show()
```

2. Partial Dependence Plots (PDP)

Explanation:

Partial dependence plots show the relationship between a specific feature and the predicted outcome while keeping other features constant. This helps understand the marginal effect of a feature on the prediction.

Visualization:

- **Line Plots or 2D Contour Plots:** For single features, line plots are used, while for two features, 2D contour plots are used.
- **Example:**

```
python
Copy code
from sklearn.inspection import plot_partial_dependence

features = ['feature1', 'feature2']  # replace with actual feature names

fig, ax = plt.subplots(figsize=(10, 6))
plot_partial_dependence(model, X_train, features, ax=ax)
plt.show()
```

Prog./Sem.: M.tech /II	Course Title: Data Science	Course Code:2MCS1-01
Duration: 1.5 hours	Date: 10,June,2024 Session : FIRST	Max Marks: 20
Submitted By: Dr. Meenakshi Nawal		

3. SHAP Values (SHapley Additive exPlanations)

Explanation:

SHAP values provide a unified measure of feature importance for individual predictions by calculating the contribution of each feature to the difference between the actual prediction and the mean prediction. This method is based on cooperative game theory.

Visualization:

- **Summary Plots:** Show the distribution of SHAP values for all features.
- **Dependence Plots:** Show the effect of a single feature across the dataset.
- **Force Plots:** Explain individual predictions.
- **Example:**

```
python
Copy code
import shap

# Create a SHAP explainer object
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_train)

# Summary Plot
shap.summary_plot(shap_values, X_train)

# Dependence Plot for a single feature
shap.dependence_plot('feature1', shap_values, X_train)

# Force Plot for an individual prediction
shap.initjs()
shap.force_plot(explainer.expected_value, shap_values[0, :], X_train.iloc[0, :])
```

Additional Techniques:

4. LIME (Local Interpretable Model-agnostic Explanations)

Explanation:

LIME explains individual predictions by approximating the model locally with an interpretable model (like linear regression). It helps understand the impact of features on a specific prediction.

Visualization:

- **Explanation Plots:** Highlight the contribution of each feature to a single prediction.
- **Example:**

```
python
Copy code
import lime
from lime import lime_tabular

explainer = lime_tabular.LimeTabularExplainer(X_train.values,
feature_names=features, class_names=['class1', 'class2'], mode='classification')
exp = explainer.explain_instance(X_test.iloc[0], model.predict_proba)
exp.show_in_notebook(show_table=True)
```

Summary

Visualizing the interpretability of machine learning models using techniques like feature importance plots, partial dependence plots, SHAP values, and LIME can provide deep insights into model behavior, feature impact, and individual predictions. These techniques help demystify complex models, making them more transparent and trustworthy.

Solution 6 : Data Science Project: Predicting Customer Churn in a Telecommunications Company

Objectives:

The primary objective of this project was to predict customer churn for a telecommunications company. By identifying customers who are likely to leave, the company aimed to implement targeted retention strategies to reduce churn rates and increase customer retention.

Methodologies Used:

1. **Data Collection:**
   - Data was collected from the company's customer database, which included customer demographics, account information, usage patterns, customer service interactions, and previous churn history.
2. **Data Preprocessing:**
   - **Data Cleaning:** Handled missing values through imputation, removed duplicates, and corrected data entry errors.
   - **Feature Engineering:** Created new features such as average monthly charges, tenure in months, and interaction frequency. Categorical variables were encoded using one-hot encoding.
   - **Normalization and Scaling:** Applied MinMaxScaler to scale numerical features to a range of 0 to 1.
3. **Exploratory Data Analysis (EDA):**

- **Univariate Analysis:** Examined the distribution of individual features.
- **Bivariate Analysis:** Explored relationships between features and the target variable (churn).
- **Correlation Analysis:** Identified correlations between features to detect multicollinearity.

4. **Model Selection and Training:**
   - **Train-Test Split:** Split the data into training (70%) and testing (30%) sets.
   - **Model Selection:** Evaluated several models including Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, and Support Vector Machines.
   - **Hyperparameter Tuning:** Used GridSearchCV to optimize hyperparameters for the best-performing models.
5. **Model Evaluation:**
   - **Performance Metrics:** Evaluated models using accuracy, precision, recall, F1-score, and AUC-ROC curve.
   - **Cross-Validation:** Applied k-fold cross-validation to ensure model robustness and generalization.
6. **Model Interpretation:**
   - **Feature Importance:** Used feature importance plots to identify key drivers of churn.
   - **Partial Dependence Plots:** Analyzed the marginal effect of key features on churn probability.
   - **SHAP Values:** Employed SHAP values to explain individual predictions and overall feature impact.
7. **Deployment:**
   - **Model Deployment:** Implemented the final model in a production environment using a web application interface.
   - **Integration with CRM:** Integrated the churn prediction model with the company's CRM system to provide real-time churn risk scores for customer service representatives.

Outcomes:

1. **Improved Churn Prediction:**
   - The final model achieved an accuracy of 87%, with a recall of 85% and an AUC-ROC score of 0.91, indicating strong performance in identifying customers likely to churn.
2. **Insightful Discoveries:**
   - Key features influencing churn included contract type, monthly charges, tenure, and customer service interactions. Customers with month-to-month contracts and higher monthly charges were more likely to churn.
3. **Actionable Recommendations:**
   - The company implemented targeted retention strategies such as offering discounts or personalized services to high-risk customers identified by the model.
   - Improved customer service protocols to proactively address issues for customers flagged as high churn risk.
4. **Business Impact:**
   - The implementation of the churn prediction model and subsequent retention strategies led to a 15% reduction in churn rates over six months, resulting in significant cost savings and increased customer lifetime value.
5. **Enhanced Customer Experience:**
   - By addressing customer issues proactively and offering tailored solutions, the company improved

overall customer satisfaction and loyalty.

This project demonstrated the power of data science in solving real-world business problems and provided the telecommunications company with a valuable tool for managing customer churn effectively.

Solution 7 : Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. It is a powerful means of communicating complex data insights in a clear and concise manner.

Importance of Data Visualization in Data Analysis

1. **Simplifies Complex Data:**
   - Converts large and complex datasets into understandable visual formats, making it easier to grasp the insights and trends.
2. **Identifies Patterns and Trends:**
   - Helps in recognizing patterns, correlations, and trends that might not be apparent in raw data.
3. **Facilitates Better Decision-Making:**
   - Provides a visual context that helps stakeholders and decision-makers quickly understand insights and make informed decisions.
4. **Enhances Communication:**
   - Simplifies the communication of data findings to non-technical stakeholders through intuitive visual representations.
5. **Detects Anomalies:**
   - Helps in identifying outliers and anomalies which can indicate errors or significant findings that need further investigation.
6. **Improves Data Accessibility:**
   - Makes data more accessible to a broader audience, enabling more people to engage with and understand the data.

Common Tools and Libraries for Data Visualization

1. Python Libraries:

- **Matplotlib:**
  - A comprehensive library for creating static, animated, and interactive visualizations in Python.
  - Example:

```
python
Copy code
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4], [10, 20, 25, 30])
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
```

```
plt.title('Simple Line Plot')
plt.show()
```

- **Seaborn:**
  - Built on top of Matplotlib, it provides a high-level interface for drawing attractive and informative statistical graphics.
  - Example:

```
python
Copy code
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(style="darkgrid")
data = sns.load_dataset("iris")
sns.scatterplot(x="sepal_length", y="sepal_width", hue="species", data=data)
plt.show()
```

- **Plotly:**
  - An interactive graphing library that makes it easy to create interactive plots and dashboards.
  - Example:

```
python
Copy code
import plotly.express as px

data = px.data.iris()
fig = px.scatter(data, x='sepal_width', y='sepal_length', color='species',
title='Iris Dataset')
fig.show()
```

- **Bokeh:**
  - A powerful library for creating interactive visualizations for modern web browsers.
  - Example:

```
python
Copy code
from bokeh.plotting import figure, output_file, show

output_file("line.html")
p = figure(title="Simple Line Example")
p.line([1, 2, 3, 4], [3, 7, 8, 5], line_width=2)
show(p)
```

- **Altair:**
  - A declarative statistical visualization library based on Vega and Vega-Lite that enables the creation of interactive and meaningful visualizations with concise syntax.
  - Example:

```
python
Copy code
import altair as alt
import pandas as pd

data = pd.DataFrame({
    'x': [1, 2, 3, 4, 5],
    'y': [2, 3, 4, 5, 6]
})

chart = alt.Chart(data).mark_line().encode(
    x='x',
    y='y'
)
chart.show()
```

2. R Libraries:

- **ggplot2:**
  - A data visualization package for the R programming language that is based on the grammar of graphics.
  - Example:

```
R
Copy code
library(ggplot2)
data <- data.frame(x = c(1, 2, 3, 4), y = c(2, 3, 5, 7))
ggplot(data, aes(x = x, y = y)) + geom_line() + ggtitle('Simple Line Plot')
```

- **Shiny:**
  - A web application framework for R that allows the creation of interactive web applications with R.
  - Example:

```
R
Copy code
library(shiny)

ui <- fluidPage(
  titlePanel("Hello Shiny!"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins", "Number of bins:", min = 1, max = 50, value = 30)
    ),
    mainPanel(
      plotOutput("distPlot")
    )
  )
)

server <- function(input, output) {
```

Solution of Question Paper
II Mid-Term Examination, June -2024

Prog./Sem.: M.tech /II	**Course Title: Data Science**	**Course Code:2MCS1-01**
Duration: 1.5 hours	**Date: 10,June,2024** **Session : FIRST**	**Max Marks: 20**
Submitted By: Dr. Meenakshi Nawal		

```
  output$distPlot <- renderPlot({
    x <- faithful$eruptions
    bins <- seq(min(x), max(x), length.out = input$bins + 1)
    hist(x, breaks = bins, col = 'darkgray', border = 'white')
  })
}

shinyApp(ui = ui, server = server)
```

Summary

Data visualization is a critical component of data analysis, enabling the simplification, communication, and effective interpretation of data insights. It helps in identifying patterns, making informed decisions, and enhancing the accessibility of data. Tools and libraries like Matplotlib, Seaborn, Plotly, Bokeh, Altair (for Python), and ggplot2, Shiny (for R) are widely used to create a variety of visualizations that facilitate the understanding and presentation of data.

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur		
M.Tech_ I Year_ II Semester (Session 2023-2024)		
CO Attainment		
Name of the Exam: Theory Mid term -II	Department:CSE	Section: N.A
Faculty Name: Dr. Meenakshi Nawal	Course Title with CODE: Data Science, 2MCS1-01	
CO for the course: **Upon successful completion of this course, students will be able to:**		
CO1: Building the fundamentals of data science.		
CO2: Imparting design thinking capability to build big-data.		
CO3: Developing design skills of models for big data problems .		
CO4: Gaining practical experience in programming tools for data sciences.		
CO5: Empowering students with tools and techniques used in data science.		

MID TERM EVALUATION

S. NO.	ROLL NO	PART →	A			B		C	Total
		QUESTION NO. →	Q1	Q2	Q3	Q4	Q5	Q6/Q7	
		COURSE OUTCOME(S) SATISFIED →	CO5	CO5	CO5	CO4	CO3	CO5	
		MAXIMUM MARKS →	2	2	2	4	4	6	
		MINIMUM QUALIFYING MARKS (60%) →	1.2	1.2	1.2	2.4	2.4	3.6	
		NAME OF STUDENT ↓							
1	23ESKCS600	Kritika Sharma	2	1	2	3	2.5	6	17
2	23ESKCS601	Muzamil Riyaz	AB	AB	AB	AB	AB	AB	AB
Total Students in Class (S1)			2	2	2	2	2	2	
No. of Students scored >=60% marks (S2)			1	0	1	1	1	1	
Weighted marks as per Total Students in Class (W_{avg} of S1)			4	4	4	8	8	12	
Weighted marks as per No. of Students scored >=60% marks (W_{avg} of S2)			2	0	2	4	4	6	

CO Attainment Level		Weighted Avg. Percentage of CO Attainment (N)	CO Attainment Level
Attainment of	CO1	NIL	NIL
Attainment of	CO2	NIL	NIL
Attainment of	CO3	50	1
Attainment of	CO4	50	1
Attainment of	CO5	41.67	1
Criterion of Percentage for CO Attainment Level considering target 60%		0 < N ≤ 50%	Low (1)
		50% < N ≤ 70%	Medium (2)
		70% < N ≤ 100%	High (3)

Weighted Avg. percentage attainment of CO_j :

$$CO_j = \frac{\sum_{i=1}^{n} S2_i \times MM_i}{\sum_{i=1}^{n} S1_i \times MM_i} \times 100$$

where i = *Index of question number correlated with* CO_j

j = *Index of the course outcome*

n = *Number of questions*

MM = *Maximum marks*

Nawal 11/06/2024

Dr. Meenakshi Nawal

Faculty name with signature

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

M.Tech I Year II Semester (Session 2023-2024)

CO Attainment

Name of the Exam: Theory Mid term -I	**Department: CSE**	**Section:**
Faculty Name: Dr. Nilam	**Course Title with CODE: DPA (2MCS1-02)**	

CO for the course:
Upon successful completion of this course, students will be able to:

CO1: Identify the concepts of parallel processing and understand the particular problems arising in the programming of parallel machines

CO2: Compare and Evaluate the parallel computing models and the "parallel-way of thinking" required in the design of parallel algorithms

CO3: Apply the basic algorithmic techniques and design algorithms in shared memory as well as distributed memory environment

CO4: Acquire knowledge of those problems which have been perceived as intractable for parallelization

CO5: Understand and be able to apply basic parallel programming principles in a shared/distributed memory environment

CO6: Classify the strengths and limitations of parallel computing approaches to problem-solving

MID TERM EVALUATION

S. NO.	ROLL NO	PART →	A			B		C	Total
		QUESTION NO. →	Q1	Q2	Q3	Q4	Q5	Q6/Q7	
		COURSE OUTCOME(S) SATISFIED →	CO3	CO4	CO5	CO5	CO3	CO6	
		MAXIMUM MARKS →	2	2	2	4	4	6	
		MINIMUM QUALIFYING MARKS (60%) →	1.2	1.2	1.2	2.4	2.4	3.6	
		NAME OF STUDENT ↓							
1	23ESKCS600	Kritika Sharma	2	2	0	4	4	6	19
2	23ESKCS601	Muzamil Riyaz	AB	AB	AB	AB	AB	AB	AB
Total Students in Class (S1)			2	2	2	2	2	2	
No. of Students scored >=60% marks (S2)			1	1	0	1	1	1	

CO Attainment Level	**Weighted Avg. Percentage of CO Attainment (N)**	**CO Attainment Level**
Attainment of CO-1	-	-
Attainment of CO-2	-	-
Attainment of CO-3	**100.00%**	**3**
Attainment of CO-4	**50%**	**1**
Attainment of CO-5	**50%**	**1**
Attainment of CO-6	**50%**	**1**
Criterion of Percentage for CO Attainment Level considering target 60%	$0 < N \leq 50\%$	Low (1)
	$50\% < N \leq 70\%$	Medium (2)
	$70\% < N \leq 100\%$	High (3)

Faculty name with signature Dr. Nilam

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

II Mid Term Examination, Dec.- 2023

MBA./ Semester III | **Branch: OSM**
Subject: Total Quality Management | **Subject Code : M-341**
Time: 1½ Hours | **Maximum Marks : 20**

PART A: **(3 X 2=6 Marks)**

Attempt all questions. Each question carries 2 Marks.

Q1. Explain the significance of brainstorming and affinity diagram.

Q2. What are variable control charts and attribute control charts? What are the advantages of control charts?

Q3. What are the objectives of ISO 14000 series?

PART B: **(2 X 4 = 8 Marks)**

Attempt any 2 questions out of 3 questions. Each question carries 4 Marks.

Q4. Explain the significance of Check sheet and Cause-effect diagram for quality improvement with the help of suitable examples.

Q5. What is Quality Function Deployment? Explain the matrices involved in QFD process.

Q6. What is the Taguchi's quality loss function? Explain 'signal to noise ratio' with suitable example?

PART C: **(1 X 6= 6 Marks).**

Q7. 'The goal of benchmarking is to identify best practices, understand performance gaps, and implement improvements to enhance overall efficiency and effectiveness.' Describe how benchmarking can be used to improve both efficiency and effectiveness.

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Analysis of Question Paper

III Sem MBA II Mid-Term Examination, 2023-24

Branch/Semester: MBA-III	**Subject:** TQM	**Subject Code: M-341**
Duration: 1.5 hours	**Session (I/II/III): I**	**Max Marks: 20**
Submitted By:	**Dr. Deepak Kumar**	

A. Distribution of Course Outcome and Bloom's Taxonomy in Question Paper

Q. No	Questions	Marks	CO	BL
1	Explain the significance of brainstorming and affinity diagram.	2	3	2
2	What are variable control charts and attribute control charts? What are the advantages of control charts?	2	4	2
3	What are the objectives of ISO 14000 series?	2	3	2
4	Explain the significance of Check sheet and Cause-effect diagram for quality improvement with the help of suitable examples.	4	3	3
5	What is Quality Function Deployment? Explain the matrices involved in QFD process.	4	3	3
6	What is the Taguchi's quality loss function? Explain 'signal to noise ratio' with suitable example?	4	5	3
7	'The goal of benchmarking is to identify best practices, understand performance gaps, and implement improvements to enhance overall efficiency and effectiveness.' Describe how benchmarking can be used to improve both efficiency and effectiveness.	6	4	4

BL – Bloom's Taxonomy Level
(1- Remembering, 2- Understanding, 3 – Applying, 4 – Analyzing, 5 – Evaluating, 6 - Creating)
CO – Course Outcome

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Analysis of Question Paper

III Sem MBA II Mid-Term Examination, 2023-24

Branch/Semester: MBA-III	Subject: TQM	Subject Code: M-341
Duration: 1.5 hours	Session (I/II/III): I	Max Marks: 20
Submitted By:	Dr. Deepak Kumar	

B. Questions and Course Outcomes (COs) Mapping in terms of correlation

COs	Q1	Q2	Q3	Q4	Q5	Q6	Q7
CO1	-	-	-	-	-	-	-
CO2	-	-	-	-	-	-	-
CO3	3	-	2	3	3	-	-
CO4	-	2	-	-	-	-	3
CO5	-	-	-	-	-	3	-

1-Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation

C. Mapping of Bloom's Level and Course Outcomes with Question Paper

Bloom's Level Mapping		CO Mapping	
Bloom's Level	Percentage	CO	Percentage
BL1	-	CO1	-
BL2	43	CO2	-
BL3	43	CO3	57
BL4	14	CO4	28
BL5	-	CO5	15
BL6	-	-	-

Swami Keshvanand Institute of Technology, Management &Gramothan, Jaipur

II Mid Term Examination, Dec.-2023

Semester:	**III**	**Branch:**	**MBA(OSM)**
Subject:	**Manufacturing planning and control**	**Subject Code:**	**M-342**
Time:	**1.5 Hours**	**Maximum Marks:**	**20**
Session (I/II/III):II			

PART A (short-answer type questions)

(All questions are compulsory) (3*2=6)

Q1. Explain Kanban card system.

Q2. Explain chase and level strategies of operations management.

Q3. Explain Just in time manufacturing and Material requirement planning.

PART B (Analytical/Problem solving questions)

(Attempt any 2 Questions) (2*4=8)

Q4. Draw the Framework for Production Activity Control.

Q5. Explain the hierarchy of capacity planning.

Q6. Explain the scope of ERP applications in manufacturing planning and control.

PART C (Descriptive/Analytical/Problem solving/Design questions)

(Attempt any 1 Question) (1*6=6)

Q7. Mehta Tools has to schedule seven jobs on a single machine. Using the information presented in the following table, identifythe processing sequence that would result using (1) FCFS, and (2) EDD. Jobs are listed in order of arrival. For eachmethod, determine:

(i) Average job flow time

(ii) Average job tardiness

Job	A	B	C	D	E	F	G
Processing Time (Days)	4	12	2	11	10	3	6
Due Date (Days)	20	30	15	16	18	5	9

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Analysis of Question Paper

II Mid-Term Examination, Dec.- 2023

Branch/Semester:III	**Subject: Manufacturing Planning and Control**	**Subject Code: M-342**
Duration: 1.5 hours	**Session (I/II/III): II**	**Max Marks:20**
Submitted By: Dr. D.Hariyani		

A. Distribution of Course Outcome and Bloom's Taxonomy in Question Paper

Q. No	Questions	Marks	CO	BL
1	Explain Kanban card system.	2	5	2
2	Explain chase and level strategies of operations management.	2	2	2
3	Explain Just in time manufacturing and Material requirement planning.	2	2	2
4	Draw the Framework for Production Activity Control.	4	5	2
5	Explain the hierarchy of capacity planning.	4	4	2
6	Explain the scope of ERP applications in manufacturing planning and control.	4	5	2
7	Mehta Tools has to schedule seven jobs on a single machine. Using the information presented in the following table, identify the processing sequence that would result using (1) FCFS, and (2) EDD. Jobs are listed in order of arrival. For each method, determine: (i) Average job flow time (ii) Average job tardiness	6	5	3
8	Continental Paint Works have to process seven jobs, whereeach job has to pass through two machines (M1 and M2). Estimates of processing times (in mins) of the jobs on the Machine M1 and M2 are listed below: (i) Apply Johnson's algorithm and determine processing sequence that minimizes the flow time. (ii) Develop a Gantt chart for scheduling the jobs based on the sequence obtained from part (i).	6	5	3

Job	A	B	C	D	E	F	G
Processing Time (Days)	4	12	2	11	10	3	6
Due Date (Days)	20	30	15	16	18	5	9

Job	A	B	C	D	E	F	G
Machine 1	5	4	8	12	9	2	10
Machine 2	6	3	9	15	8	7	11

BL – Bloom's Taxonomy Level
(1- Remembering, 2- Understanding, 3 – Applying, 4 – Analyzing, 5 – Evaluating, 6 - Creating)
CO – Course Outcome

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

Analysis of Question Paper

II Mid-Term Examination, Dec.- 2023

Branch/Semester:III	**Subject: Manufacturing Planning and Control**	**Subject Code: M-342**
Duration: 1.5 hours	**Session (I/II/III): II**	**Max Marks:20**
Submitted By: Dr. D.Hariyani		

B. Questions and Course Outcomes (COs) Mapping in terms of correlation

COs	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
CO1	2							
CO2		2						
CO3			2					
CO4					2			
CO5				3		3	3	3

1-Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation

C. Mapping of Bloom's Level and Course Outcomes with Question Paper

Bloom's Level Mapping		CO Mapping	
Bloom's Level	Percentage	CO	Percentage
BL1		CO1	12.5
BL2	75	CO2	12.5
BL3	25	CO3	12.5
BL4	-	CO4	12.5
BL5	-	CO5	50
BL6	-	CO6	-

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

II Mid Term Examination, Dec.- 2023

MBA./ Semester – III | **Branch: OSM**
Subject Total Quality Management | **Subject Code : M-341**
Time: 1½ Hours | **Maximum Marks : 20**

PART A: **(3 X 2=6 Marks)**

Attempt all questions. Each question carries 2 Marks.

Q1. Explain the significance of brainstorming and affinity diagram.

Q2. What are variable control charts and attribute control charts? What are the advantages of control charts?

Q3. What are the objectives of ISO 14000 series?

PART B: **(2 X 4 = 8 Marks)**

Attempt any 2 questions out of 3 questions. Each question carries 4 Marks.

Q4. Explain the significance of Check sheet and Cause-effect diagram for quality improvement with the help of suitable examples.

Q5. What is Quality Function Deployment? Explain the matrices involved in QFD process.

Q6. What is the Taguchi's quality loss function? Explain 'signal to noise ratio' with suitable example?

PART C: **(1 X 6= 6 Marks).**

Q7. 'The goal of benchmarking is to identify best practices, understand performance gaps, and implement improvements to enhance overall efficiency and effectiveness.' Describe how benchmarking can be used to improve both efficiency and effectiveness.

Swami Keshvanand Institute of Technology, Management &Gramothan,
Ramnagaria, Jagatpura, Jaipur-302017

Question Paper Solution

Branch : MBA **Semester: III** **Subject: TQM M-341** **Mid Term: II**

Submitted By : Dr. Deepak Kumar.

Ans 1	Brainstorming is a creative group problem-solving technique where participants generate a large number of ideas spontaneously. It encourages open and free-thinking, aiming to explore diverse possibilities without immediate evaluation or criticism. Affinity diagrams- Organizing Information: Affinity diagrams help in organizing and structuring large amounts of information gathered during quality improvement initiatives. This structured organization facilitates a better understanding of the issues and challenges faced by the organization. Problem Identification: By grouping related ideas or issues together, the affinity diagram assists in identifying common themes or patterns contributing to quality problems. This is essential for addressing root causes rather than just symptoms.
Ans 2	Variable control charts and attribute control charts are tools used in statistical process control (SPC) to monitor and control the quality of a process. They are part of a broader set of techniques aimed at maintaining and improving process stability and consistency. 1. **Variable Control Charts:** • **Definition:** Variable control charts are used when the quality characteristics being measured are quantitative and can be represented by numerical data (e.g., dimensions, weight, temperature). • **Example:** A variable control chart might be used to monitor the thickness of a coating on a product. (X bar and R chart) • **Advantages:** They provide a graphical representation of how a process variable changes over time, helping to identify trends, shifts, and patterns. This allows for the early detection of deviations from the desired quality standards. 2. **Attribute Control Charts:** • **Definition:** Attribute control charts are used when the quality characteristic being monitored is qualitative and can be categorized into discrete categories (e.g., pass/fail, good/bad). • **Example:** An attribute control chart might be used to monitor the proportion of defective items in a production batch. • **Advantages:** They are useful for monitoring the proportion of nonconforming items, making them simpler to use for certain types of data. Attribute control charts are particularly helpful when dealing with binary outcomes. **Advantages of Control Charts:** 1. **Early Detection of Issues:** Control charts enable the early detection of variations and trends in a process, allowing for timely intervention and correction before defects become widespread. 2. **Process Understanding:** By plotting data points over time, control charts provide a visual representation of the stability and capability of a process. This aids in understanding the inherent variability and performance of the process. 3. **Objective Decision-Making:** Control charts offer a systematic and objective basis for making decisions about process stability. They provide statistical criteria for determining whether observed variations are due to common causes (inherent to the process) or special causes (indicating a problem that needs attention). 4. **Continuous Improvement:** Control charts are integral to the philosophy of continuous improvement in quality management. They support ongoing efforts to identify and eliminate sources of variability, leading to sustained improvements in process performance.

Question Paper Solution

Branch : MBA **Semester: III** **Subject: TQM M-341** **Mid Term: II**

Submitted By : Dr. Deepak Kumar.

Ans 3	Environmental Management System (EMS) Implementation: Establishing and implementing an effective environmental management system based on the ISO 14001 standard. This involves developing policies, procedures, and practices to systematically manage environmental aspects and impacts. Compliance with Legal and Regulatory Requirements: Ensuring that organizations comply with applicable environmental laws, regulations, and other requirements. The standards help organizations identify and understand the legal obligations relevant to their environmental aspects. Risk Assessment and Management: Identifying and assessing environmental aspects and impacts associated with an organization's activities, products, and services. This includes evaluating potential risks and implementing measures to manage and mitigate these risks.
Ans . 4	Significance: A check sheet is a simple and effective tool for systematically collecting and organizing data. It is used to identify patterns, frequencies, and trends in a process or a set of activities. Significantly aids in data collection and analysis during the initial stages of quality improvement projects. Example: Imagine a manufacturing process where defects in the final product need to be identified and categorized. A check sheet can be used by quality inspectors to record the types and frequencies of defects observed during a specific timeframe. The check sheet might have categories such as "Scratches," "Incomplete Assembly," and "Mismatched Parts." Each time a defect is found, a mark or tally is made in the corresponding category on the check sheet. Significance in Quality Improvement: Helps visualize and quantify the types and frequency of issues, facilitating a data-driven approach to problem-solving. Enables teams to prioritize issues based on the most common or critical problems identified. Guides decision-making by providing a clear picture of where improvements are needed. Cause-and-Effect Diagram (Fishbone Diagram): Significance: The Cause-and-Effect Diagram, also known as the Fishbone Diagram, is a visual tool used to identify and explore potential causes of a specific problem. It encourages systematic thinking about the various factors that may contribute to an issue. Significantly aids in root cause analysis and problem-solving by organizing and categorizing potential causes.

	Example: Consider a scenario where a manufacturing process is experiencing a higher-than-acceptable defect rate. The Fishbone Diagram can be used to identify and categorize potential causes contributing to this problem. Categories on the diagram might include "Manpower," "Methods," "Materials," "Machines," "Environment," and "Measurement." Within each category, specific factors that could contribute to defects are identified. Significance in Quality Improvement: Promotes a structured and comprehensive analysis of potential causes, preventing the overlooking of critical factors. Encourages cross-functional collaboration as team members from different areas contribute their insights to the diagram.
Ans 5)	Quality Function Deployment (QFD) is a process and set of tools used to effectively define customer requirements and convert them into detailed engineering specifications and plans to produce the products that fulfill those requirements. QFD is used to translate customer requirements (or VOC) into measureable design targets and drive them from the assembly level down through the sub-assembly, component and production process levels. QFD methodology provides a defined set of matrices utilized to facilitate this progression. Step 1: Customer Requirements – "Voice of the Customer" ... Step 2: Regulatory Requirements. ... Step 3: Customer Importance Ratings. ... Step 4: Customer Rating of the Competition. ... Step 5: Technical Descriptors – "Voice of the Engineer" ... Step 6: Direction of Improvement. ... Step 7: Relationship Matrix.
Ans 6	Taguchi's Quality Loss Function, developed by Genichi Taguchi, is a concept in quality management that quantifies the economic loss to society as a result of variation in product or process characteristics. The fundamental idea is that deviations from the target or optimal values for these characteristics result in increased costs and decreased customer satisfaction. The Quality Loss Function is expressed mathematically as: $2L(y)=k\times(y-T)2$ Where: • (◆)$L(y)$ is the loss incurred by society. • k is a constant that represents the cost of poor quality (a constant multiple). • y is the deviation from the target or optimal value. • T is the target or nominal value.

The goal is to minimize the total loss to society by minimizing the deviation from the target value.

Signal-to-Noise Ratio (S/N Ratio):

In the context of Taguchi's methods, the Signal-to-Noise Ratio (S/N Ratio) is a measure used to evaluate the performance of a product or process in the presence of variability. It is a ratio of the "signal," which represents the mean performance, to the "noise," which represents the variability or deviation from the target.

The formula for the S/N Ratio is often expressed as:

=−10×log⁡10(mean of squared deviations from targettarget variance)S/N=−10×log10

(target variancemean of squared deviations from target)

A higher S/N Ratio indicates better performance because it signifies a higher signal (mean) relative to the noise (variability).

Example: Let's say we are manufacturing a product, and the target dimension for a critical feature is 100 units. We want to minimize variability around this target.

- If the mean dimension of the product is 100 units, and the standard deviation (representing variability) is 2 units, the S/N Ratio would be calculated as:
  =−10×log⁡10(2222)=−10×log⁡10(1)=0S/N=−10×log10(2222)=−10×log10(1)=0
- If the mean dimension is still 100 units, but the standard deviation decreases to 1 unit, the S/N Ratio would be: =−10×log⁡10(1212)=−10×log⁡10(1)=0S/N=−10×log10(1212)=−10×log10(1)=0

In both cases, the S/N Ratio is 0, indicating that the mean is equal to the noise. However, the second scenario, with lower variability, is considered more desirable in terms of quality. The goal is to maximize the S/N Ratio, reflecting a situation where the signal (mean) is much larger than the noise (variability).

Ans 7

Benchmarking is the practice of comparing business processes and performance metrics to industry bests and best practices from other companies. Dimensions typically measured are quality, time and cost. Benchmarking is used to measure performance using a specific indicator (cost per unit of measure, productivity per unit of measure, cycle time of x per unit of measure or defects per unit of measure) resulting in a metric of performance that is then compared to others.

- Businesses can use benchmarking in their operations to measure themselves against internal or external standards.
- Benchmarking can be used to measure internal progress, performance against competitors and how our processes rank against world-class organizations.
- The process of benchmarking involves identifying key metrics, assessing where you currently are, determining where you want to be and creating an action plan to achieve and measure our goals.

a) Internal benchmarking: Internal benchmarking is all about improving our business by comparing it to historical data. Whether you're comparing organizational departments or different branch locations, you can use internal benchmarking to uncover the best, most efficient practices and share them across the company.

b) Competitive benchmarking : As the name suggests, competitive benchmarking is about setting certain goals based on what our competitors are doing. By studying the practices and standards of similar businesses to match or, ideally, exceed the industry status quo, our business can gain a competitive edge.

c) Strategic benchmarking : One step beyond competitive benchmarking is strategic benchmarking, in which a business seeks to emulate specific performance standards of world-class organizations. This may involve cross-industry inspiration, like when Southwest Airlines modeled its maintenance, cleaning and boarding processes after the time-bound, defined tasks of a well-oiled NASCAR pit crew.

Swami Keshvanand Institute of Technology, Management &Gramothan, Jaipur

II Mid Term Examination, Dec.-2023

Semester:	**III**	**Branch:**	**MBA(OSM)**
Subject:	**Manufacturing planning and control**	**Subject Code:**	**M-342**
Time:	**1.5 Hours**	**Maximum Marks:**	**20**
Session (I/II/III):II			

PART A (short-answer type questions)

(All questions are compulsory) **(3*2=6)**

Q1. Explain Kanban card system.

Q2. Explain chase and level strategies of operations management.

Q3. Explain Just in time manufacturing and Material requirement planning.

PART B (Analytical/Problem solving questions)

(Attempt any 2 Questions) **(2*4=8)**

Q4 Draw the Framework for Production Activity Control.

Q5. Explain the hierarchy of capacity planning.

Q6. Explain the scope of ERP applications in manufacturing planning and control.

PART C (Descriptive/Analytical/Problem solving/Design questions)

(Attempt any 1 Question) **(1*6=6)**

Q7. Mehta Tools has to schedule seven jobs on a single machine. Using the information presented in the following table, identifythe processing sequence that would result using (1) FCFS, and (2) EDD. Jobs are listed in order of arrival. For eachmethod, determine:

(i) Average job flow time

(ii) Average job tardiness

Job	A	B	C	D	E	F	G
Processing Time (Days)	4	12	2	11	10	3	6
Due Date (Days)	20	30	15	16	18	5	9

Swami Keshvanand Institute of Technology, Management & Gramothan, Ramnagaria, Jagatpura, Jaipur-302017

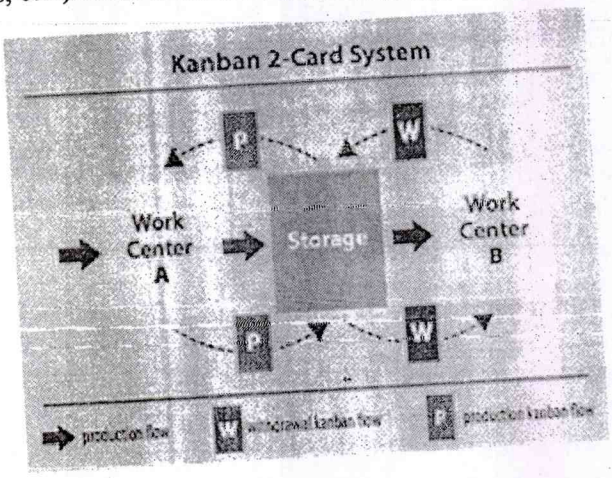
Solution of Question Paper

II Mid-Term Examination, Dec. -2023

Branch/Semester: MBA/III	**Subject: Manufacturing planning and control**	**Subject Code: M-342**
Duration: 1.5 hours	**Date: 23.12.23 Session (I/II/III): II**	**Max Marks: 20**
Submitted By: Dr. Dharmendra Hariyani		

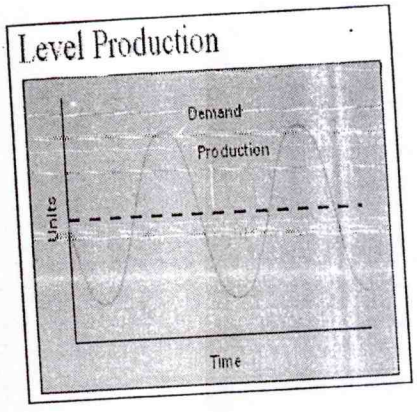
Ans 1 –

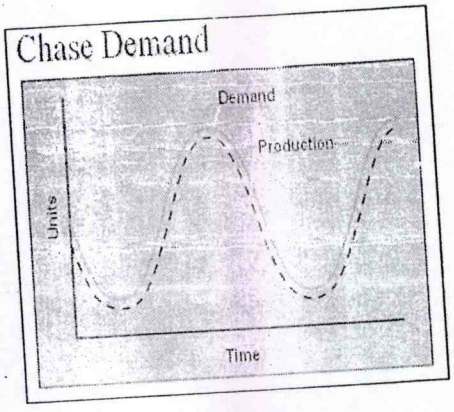
A kanban card is small card containing information about a specific part used in production. It is a signal that tells someone upstream to move, purchase, or build more of a component for production. These cards must contain the number of units (items, pounds, kits, etc.) that need action.



Ans 2 –

- **Chase Strategy** – Production output changes every time period to match the value of forecasted sales.
- **Level Strategy** – Production is at a constant rate of output with inventory buildups and depletions.





Ans 3 –

Just-In-Time (JIT) Manufacturing:

Just-In-Time (JIT) is a production strategy that focuses on producing goods or services exactly when they are needed, and in the quantity required, by the next process in the production chain. The goal of JIT is to minimize inventory levels, reduce carrying costs, and improve overall efficiency.

Key Principles:

Minimization of Inventory: JIT emphasizes the reduction of inventory levels to a minimum, which helps in

Solution of Question Paper

II Mid-Term Examination, Dec. -2023

Branch/Semester: MBA/III	Subject: Manufacturing planning and control	Subject Code: M-342
Duration: 1.5 hours	Date: 23.12.23 Session (I/II/III): II	Max Marks: 20
Submitted By: Dr. Dharmendra Hariyani		

freeing up capital and storage space.
Continuous Flow: The production process is designed for a continuous and smooth flow of materials through various stages, with minimal interruptions or delays.
Pull System: JIT operates on a pull system, where production is initiated based on customer demand. This contrasts with a push system, where production is based on forecasts or predetermined schedules.
Quality Control: Emphasis is placed on producing high-quality products from the beginning, reducing the need for rework and ensuring customer satisfaction.
Flexibility: JIT systems are designed to be flexible and responsive to changes in customer demand or market conditions.

Material Requirements Planning (MRP):

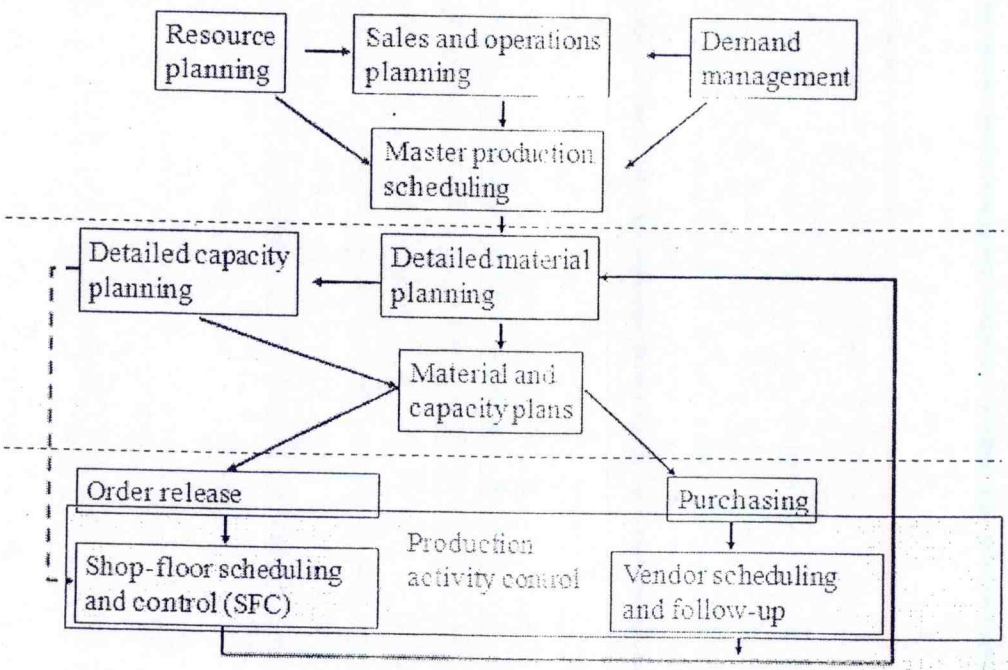
Material Requirements Planning (MRP) is a computer-based inventory management system that helps in planning and controlling the flow of materials within a manufacturing process. MRP is designed to ensure that the right materials are available at the right time to meet production demands while minimizing excess inventory.

Key Elements:

Bill of Materials (BOM): A comprehensive list of components and sub-assemblies required to manufacture a product.

Master Production Schedule (MPS): A detailed plan that specifies the quantity and timing of production for each end product.
Inventory Status: Continuous monitoring of inventory levels and updates based on production and consumption.
Ordering: Automatic generation of purchase orders or production orders to replenish materials as needed.

Ans 4 –



Solution of Question Paper

II Mid-Term Examination, Dec. -2023

Branch/Semester: MBA/III	Subject: Manufacturing planning and control	Subject Code: M-342
Duration: 1.5 hours	Date: 23.12.23 Session (I/II/III): II	Max Marks: 20
Submitted By: Dr. Dharmendra Hariyani		

Ans 5 –

Long-Range Plans

- Long – term capacity
- Location
- Layout
- Work system design
- Product design

Intermediate Plans

General levels of :

- Employment
- Output
- Finished- goods inventories
- Subcontracting
- Back orders

Short-Range Plans

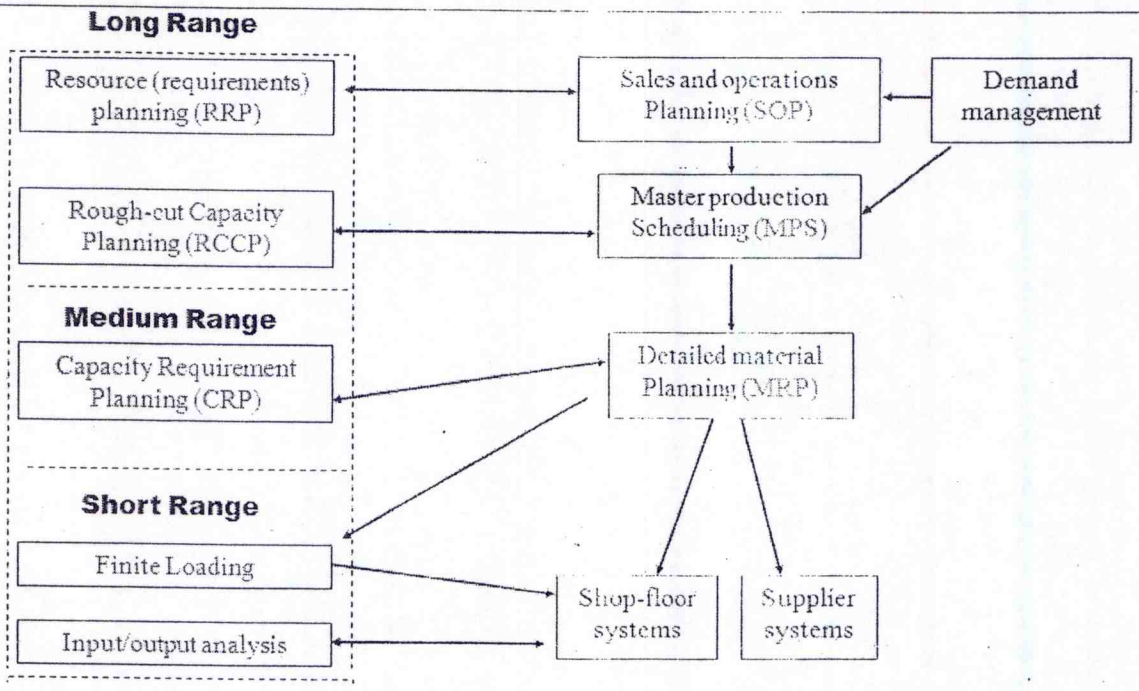
Detailed plans of :

- Production lot size
- Order quantities
- Machine loading
- Job assignments
- Job sequencing
- Work schedules

Solution of Question Paper

II Mid-Term Examination, Dec. -2023

Branch/Semester: MBA/III	Subject: Manufacturing planning and control	Subject Code: M-342
Duration: 1.5 hours	Date: 23.12.23 Session (I/II/III): II	Max Marks: 20
Submitted By: Dr. Dharmendra Hariyani		



Ans 6 - **Scope of ERP applications in manufacturing planning and control –**

- It represented and represents a comprehensive software (a data base) approach to support decisions in the enterprise (concurrent with planning and controlling the business)
- It also describes a software system that integrates application programs in many functional areas.
- The integration is made through a data base shared by all functions and data processing applications in the firm.

SAP Application Modules
1. Financial Accounting
2. Financial Supply Chain Management.
3. Controlling.
4. Material Management.
5. Sales and Distribution.
6. Logistic Execution.
7. Production Planning.
8. Quality Management.
9. Plant Maintenance.
10. Project system.

Solution of Question Paper

II Mid-Term Examination, Dec. -2023

Branch/Semester: MBA/III	Subject: Manufacturing planning and control	Subject Code: M-342
Duration: 1.5 hours	Date: 23.12.23 Session (I/II/III): II	Max Marks: 20
Submitted By: Dr. Dharmendra Hariyani		

Ans 7 – **FCFS**

Job	Processing Time (Days)	Flow Time (Days)	Due Date (Days)	Tardiness (Days)
A	4	4	20	0
B	12	16	30	0
C	2	18	15	3
D	11	29	16	13
E	10	39	18	21
F	3	42	5	37
G	6	48	9	39
7	48	196		113

Makespan = 48 days

(i) Average flow time = 196/7 = 28 days

(ii) Average Job Tardiness = 113/7 = 16.14 days

EDD

Job	Processing Time (Days)	Flow Time (Days)	Due Date (Days)	Tardiness (Days)
F	3	3	5	0
G	6	9	9	0
C	2	11	15	0
D	11	22	16	6
E	10	32	18	14
A	4	36	20	16
B	12	48	30	18
7	48	161		54

Makespan =48 days

(iii) Average flow time =161/7 – 23 days

(iv) Average Job Tardiness =54/7 – 7.71 days

Priority Rule	Average Flow Time (Days)	Average Job Tardiness (Days)	Average Number of Jobs
FCFS	28	16.14	4.08
EDD	23	7.71	3.35

Solution of Question Paper

II Mid-Term Examination, Dec. -2023

Branch/Semester: MBA/III	Subject: Manufacturing planning and control	Subject Code: M-342
Duration: 1.5 hours	Date: 23.12.23 Session (I/II/III): II	Max Marks: 20
Submitted By: Dr. Dharmendra Hariyani		

Based on above summarized results, EDD is recommended as lowest average flow (completion) time, average number of jobs are obtained using it, that results in low in process inventory. Moreover, average job tardinessis also found lowest using EDD, which results in minimizing lateness for meeting deadlines.

Ans 8:

Optimal Sequence

F	A	C	G	D	E	B

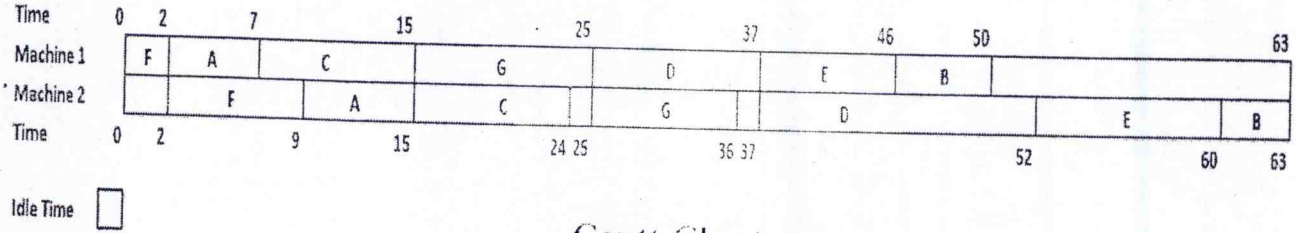
Job	Machine 1 Processing Time (mins)	Machine 2 Processing Time (mins)	Machine 1		Machine 2	
			In	Out	In	Out
F	2	7	0	2	2	9
A	5	6	2	7	9	15
C	8	9	7	15	15	24
G	10	11	15	25	25	36
D	12	15	25	37	37	52
E	9	8	37	46	52	60
B	4	3	46	50	60	63

Idle Time for Machine 1 = (63-50) = 13 mins

Idle Time for Machine 2 = (2-0) + (25-24) + (37-36) =

4 mins

Minimum Elapsed Time = 63 mins



Gantt Chart

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur	
MBA II Year III Semester (Session 2023-2024)	
CO's Attainment (Theory II Mid Term)	**Department: MBA**
Faculty Name: Dr. Deepak Kumar	**Course Name with Code: Total Quality Management (M-341)**

Upon successful completion of this course, students will be able to:

M-341.1	Understand the Basic concepts of Quality and TQM, benefits of TQM
M-341.2	Describe the scope outcome evaluation and various philosophies of quality gurus, cost of quality customer perception of quality and customer needs
M-341.3	Analyses and interpret the process quality using various as QC tools like control chart, Acceptance Sampling and apply the product quality improvement using QFD robust design and Taguchi method
M-341.4	Described as scope of benchmarking, ISO standards and manage quality programs
M-341.5	Implementation of TQM in manufacturing sector and service sector quality in design understand the significance of TQM awards

I MID TERM EVALUATION

		QUESTION NO.	1	2	3	4	5	6	7	Mid-term Total (20)
			Attempt All			Attempt any Two			Compulsory	
		COURSE OUTCOME(S) SATISFIED	3	4	3	3	3	5	4	
		MAXIMUM MARKS	2	2	2	4	4	4	6	
		MINIMUM QUALIFYING MARKS (60%)	1.2	1.2	1.2	2.4	2.4	2.4	3.6	

S.NO.	RTU Roll No.	NAME OF STUDENT								
1	22MSKXX600	Aayushi Pahariya	2	2	2		1	3	4	14
2	22MSKXX603	Amit Sharma	2	2	2		3		5	14
3	22MSKXX604	Ankit Singh	2	2	1		3	2	4	14
4	22MSKXX605	Aris Yadav	2	2	2	3	3		5	17
5	22MSKXX606	Arti Soni	2	2	2		2	2	4	14
6	22MSKXX607	Ashutosh Jangid	1	1	2		2	2	4	12
7	22MSKXX612	Deepika Babberwal	2	2	2	4		2	4	16
8	22MSKXX617	Govind Jangid	2	2	2	3	3		4	16
9	22MSKXX618	Heeralal Meena	2	2	2		4	3	4	17
10	22MSKXX624	Khushbu Bhat	2	2	1	3	3		5	16
11	22MSKXX628	Lavina Dhawan	1	1	2	3	3		4	14
12	22MSKXX633	Mukesh Kumar	1	2		4	2		4	13
13	22MSKXX636	Nihal Soni	2	2	1	3		2	4	14
14	22MSKXX641	Pranay Tiwari	2	2	2	3	4		3	16
15	22MSKXX642	Pratistha Pareek	1	2	2		3		4	12
16	22MSKXX643	Prerna Sharma	2	2	1	3		3	4	15
17	22MSKXX645	Rahul Chugh	2	2	2		4	3	4	17
18	22MSKXX646	Rashmi	1	1	2		3	3	4	14
19	22MSKXX647	Ravi Mehta	2	2	2	4		3	4	17
20	22MSKXX651	Sakshi Chaturvedi	2	2	2	4		4	4	18
21	22MSKXX655	Shrishti Kumari	1	0	0	2	0		2	5
22	22MSKXX656	Sneha Khandelwal	2	2	1	2		3	4	14
23	22MSKXX660	Vijay Motiramani	2	2	2		3	4	5	18
24	22MSKXX661	Vineet Yadav	2	2	2		4	3	3	16

	Total Students Eligible for Exam	24	24	24	24	24	24	24	
	Total Students Attempted the Question (A)	24	24	23	13	18	15	24	
	No. of Students scored >=60% marks (B)	18	20	17	11	13	10	21	
	Percentage Attainment of Criterion (B/A)	0.75	0.83	0.74	0.85	0.72	0.67	0.88	
	CO Attainment Level	3	3	3	3	3	2	3	
	Attainment of CO-3	**78%**							
	Attainment of CO-4	**67%**							
	Attainment of CO-5	**88%**							

Criterion of Paercentage for CO Attainment Level	Attainment
Percentage attainment below 60%	1
Percentage attainment 60% - 69.99%	2
Percentage attainment above and equal to 70%	3

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

MBA II Year III Semester (Session 2023-2024)

CO's Attainment (Theory I Mid Term)	Department: MBA
Faculty Name: Dr. D. Hariyani	Course Name with Code: Manufacturing Planning And Control (M-342)

Upon successful completion of this course, students will be able to:

M-342.1	Explain the demand management system in an organization for various MPC environments.
M-342.2	Explain the Sales and Operations Planning for various MPC environments.
M-342.3	Construct and develop effective Master Production Schedule Materials Requirement Plan for various MPC environments.
M-342.4	Prepare a capacity plan for various MPC environments
M-342.5	Develop an effective Production Activity Control System for various MPC environments.

I MID TERM EVALUATION

		QUESTION NO.	1	2	3	4	5	6	7	8	
			Attempt All			Attempt any Two			Compulsory		Mid-term Total (20)
		COURSE OUTCOME(S) SATISFIED	5	2	2	5	4	5	5	5	
		MAXIMUM MARKS	2	2	2	4	4	4	6	6	
		MINIMUM QUALIFYING MARKS (60%)	1.2	1.2	1.2	2.4	2.4	2.4	3.6	3.6	

S.NO.	RTU Roll No.	NAME OF STUDENT									
1	22MSKXX600	Aayushi Pahariya	1	2	2	4	4		3		16
2	22MSKXX603	Amit Sharma	1	1	2	3	2				9
3	22MSKXX604	Ankit Singh	2	2	2	3		4			13
4	22MSKXX605	Anis Yadav	1	2	2		2	4			11
5	22MSKXX606	Arti Soni	1	2	2	4		4	3		16
6	22MSKXX607	Ashutosh Jangid	1	1	1		2	2			7
7	22MSKXX612	Deepika Babberwal	1	2	2		4	4	6		19
8	22MSKXX617	Govind Jangid	2	2	2		3	4	6		19
9	22MSKXX618	Heeralal Meena	1	1	2	4		4	6		18
10	22MSKXX624	Khushbu Bhat	2	2	2		4	4	6		20
11	22MSKXX628	Lavina Dhawan	1	1	1		3	3			9
12	22MSKXX633	Mukesh Kumar									AB
13	22MSKXX636	Nihal Soni	1	2	2	2	3		3		13
14	22MSKXX641	Pranay Tiwari	2	2	2	4	4		6		20
15	22MSKXX642	Pratistha Pareek	1	1	2	4	2				10
16	22MSKXX643	Prerna Sharma	1	2	2		3	4	6		18
17	22MSKXX645	Rahul Chugh	2	2	2	3	4	4			17
18	22MSKXX646	Rashmi	1	2	2	3	3				11
19	22MSKXX647	Ravi Mehta	2	2	2	3	4		6		19
20	22MSKXX651	Sakshi Chaturvedi	2	2	2		4	4	6		20
21	22MSKXX655	Shrishti Kumari			2	1		2			5
22	22MSKXX656	Sneha Khandelwal	1	1	2		4	4	1		13
23	22MSKXX660	Vijay Motiramani	2	2	2	4	3		6		19
24	22MSKXX661	Vineet Yadav	2	2	2	4	3		6		19
	Total Students Eligible for Exam		24	24	24	24	24	24	24		
	Total Students Attempted the Question (A)		22	22	23	14	19	14	14		
	No. of Students scored >=60% marks (B)		9	16	21	12	15	12	10		
	Percentage Attainment of Criterion (B/A)		0.41	0.73	0.91	0.86	0.79	0.86	0.71		
	CO Attainment Level		1	3	3	3	3	3	3		
	Attainment of CO-2	82%									
	Attainment of CO-4	79%									
	Attainment of CO-5	71%									

Criterion of Pacrcentage for CO Attainment Level	Attainment
Percentage attainment below 60%	1
Percentage attainment 60% - 69.99%	2
Percentage attainment above and equal to 70%	3

D. Hariyani