NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 1

Q1. MCQs

(1) B. High level of communication exists between two various nodes

(2) A. Ordered

(3) D. MangoDB

(4) C. Data Node

(5) A. 4

(6) B. 2

(7) A. Cold start

(8) C. Random sampling

(9) C. Microsoft Instant Messenger.

(10) C. Periodic

Amey.

NAME: AMEY MAHENDRA THAKUR | EXAM: SEMESTER - VII | SEAT NO.: 7278000

SUBJECT: BDA | DATE: 29-11-2021 | PAGE NO.: 2

Q2 A] (i)

- The issues in stream processing mainly arise because of the following two basic reasons:
  1. The rapid rate of arrival of stream data
  2. The huge size of the data when all of the input streams are considered
- Because of the rapid arrival of stream data, the processing speed also must match the arrival speed of data. To achieve this, the entire stream processing algorithm must reside in main memory and there should be minimal secondary storage access. This is because the secondary memory accesses are many times slower that the main memory access.
- In the case of a slow stream it may be possible to process the data in that stream using the small portion of main memory. But if the number of such slow streams becomes large then again, the problem of shortage of main memory arises.
- Thus, one way to solve the issues related to stream processing is by having a large amount of main memory. But in real life compute nodes the amount of main memory available is usually limited which makes it unfeasible. So, we have to resort to some other way for solving the issues. One such way is to use the technique of sampling.

Amey

NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

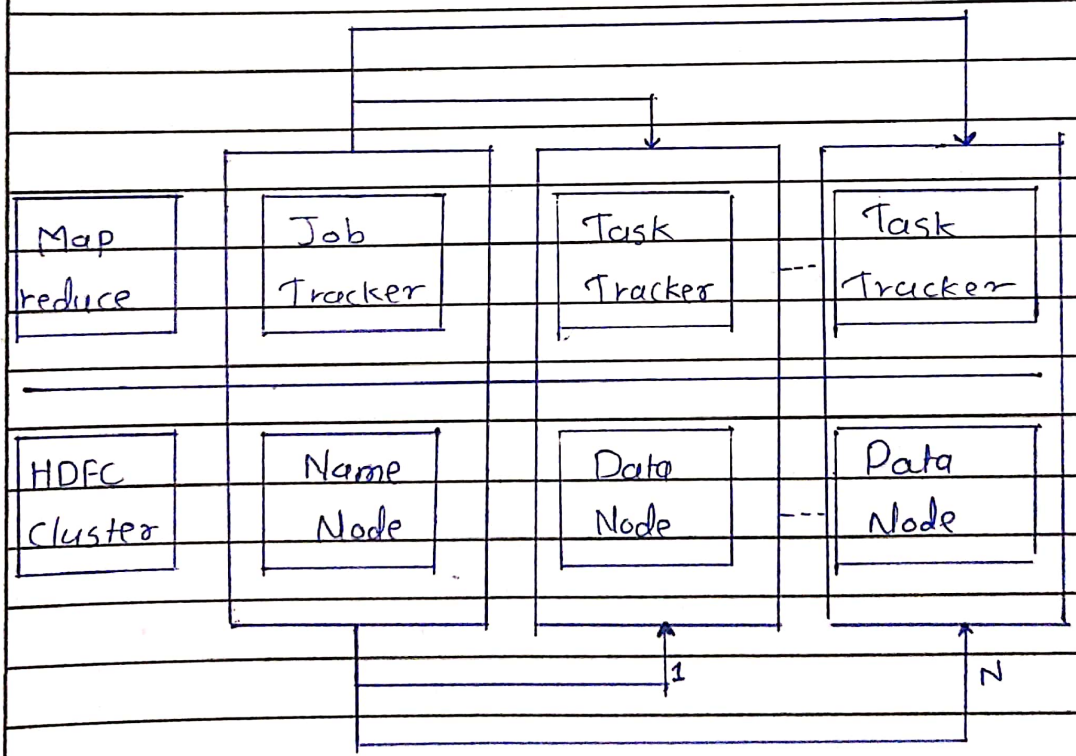
SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 3

Q2 A] - (ii)

Core Hadoop Components

- Hadoop has a master - slave topology.
- In this topology, we have one master node and multiple slave nodes.
- Master node's function is to assign a task to various slave nodes and manage resources.
- The slave node do the actual computing
- Slave nodes store the real data whereas on master we have metadata.

- Hadoop Core Components.



Amey.

Hadoop Distributed File system (HDFS)

- HDFS is a file system for hadoop
- HDFS is based on Google File System (GFS)
- It runs on clusters of commodity hardware.
- The file system have several similarities with the existing distributed file systems

Characteristics.

1. High fault tolerant.
2. High throughput
3. Support application with massive datasets.
4. Streaming access to file system data
5. Can be built out of commodity hardware

Amey

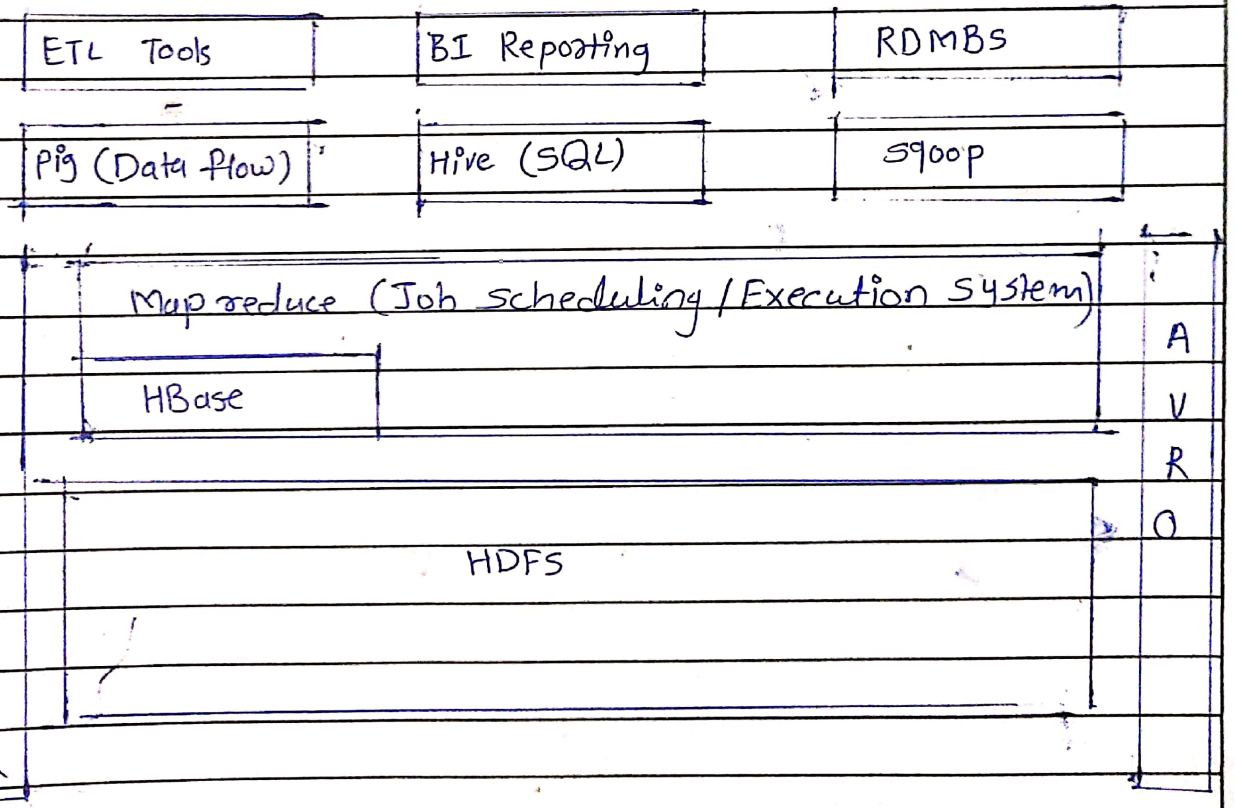
NAME: AMEY MAHENDRA THAKUR | EXAM: SEMESTER - VII | SEAT NO.: 7278000

SUBJECT: BDA | DATE: 29-11-2021 | PAGE NO.: 5

Hadoop Ecosystem Components

- Core Hadoop Ecosystem is nothing but the different components that are built on the Hadoop platform directly.
- Hadoop Ecosystem Components

1. Hadoop Distributed File System.
2. Map Reduce
3. Hive
4. Pig
5. HBase
6. Zookeeper
7. Sqoop



Amey

NAME: AMEY MAHENDRA THAKUR | EXAM: SEMESTER - VII | SEAT NO.: 7278000

SUBJECT: BDA | DATE: 29-11-2021 | PAGE NO.: 6

Q2] A] (iii)

Cosine Similarity

- Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size.
- We can measure the similarity between two sentences in Python using Cosine Similarity.
- In cosine similarity, data objects in a dataset are treated as a vector.

Formula: $\cos(x, y) = x \cdot y / \|x\| * \|y\|$.

where,

→ $x \cdot y$ = product (dot) of the vectors 'x' and 'y'.

→ $\|x\|$ and $\|y\|$ = length of the two vectors x & y.

→ $\|x\| * \|y\|$ = cross product of the two vectors 'x' and 'y'.

Amey.

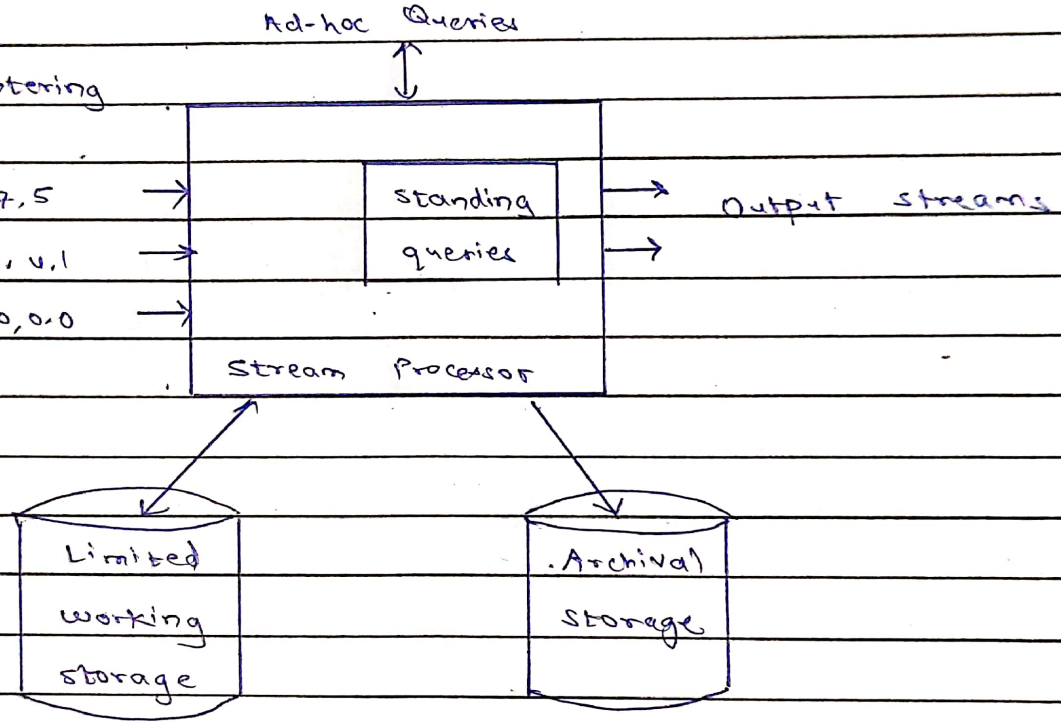
NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 7

Q2 B] (ii)

Data Stream Management System

- For handling and controlling the data streams, we require a concrete, standardized model or framework so that data stream will be processed in a properly defined manner.



Block diagram of data stream management system

- The data stream management system architecture is very much similar to that of conventional relational data base management system architecture.
- The basic difference is that processor block or more specifically a query processor (query engine) is replaced with the specialized block known as stream processor.

Amey

- The first block in the system architecture shows the input part.
- Number of data stream generated from different data sources will enter into the system.
- Every data stream has its own characteristics such as:
  1. Every data stream can schedule and rearrange its own data items.
  2. Every data stream involved heterogenity i.e. in each data stream we can find different types of data such as numeric data, alphabets, alphanumeric data, graphical data, textual data, binary data.
  3. Every data stream has different input data rate.
  4. No uniformity is maintained by the elements of different data streams while entering into the stream processor.
- In the second block of system architecture, it is abstracted that, there are two different sub systems exists one of which will take care of storing the data stream and other is responsible for fetching the data stream from secondary storage and processing it by loading into main memory.
- Hence the rate at which stream enters into the system is not the burden of sub-systems which is involved in the stream processing.
- It will be controlled by other sub-system which involved in storage of data stream.
- The third block represents the active storage or working storage for processing the different data streams.

Amey.

- The working storage area may also contain sub-systems which are integral part of main-core stream to generate result for a given query.
- Working storage basically a main memory but situation demands then data items within the stream can be fetched from the secondary storage.
- The major issue associated with working storage is its limited size.
- The fourth block of system architecture is known as archival storage.
- As name indicates, this block is responsible for maintaining the details of every transaction within the system architecture.
- It is also responsible to maintain the editlogs.
- Edit logs are nothing but updation of data.
- The fifth block is responsible for displaying or delivery the output stream generated as a result of processing done by the stream processor usually by taking the support of working storage and occasionaly by taking support of archival storage.

Amey

Q3 A] (i)

Properties of Hadoop

(1) Scalability
- Hadoop clusters are capable of scaling up and scaling down the number of nodes.

(2) Flexibility
- Hadoop cluster is very much feasible.
- It can handle any type of data irrespective of its type and structure
- It can process any type of data

(3) Speed
- Hadoop is very faster so its very efficient to work with it.
- It has fast speed because data is distributed and because data is mapped

(4) No data loss
- Data loss becomes less probable as hadoop has the ability to replicate the data
- In case of failure, data is not lost as it keeps track of backup for that data

Amey

NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 11

Limitations of Hadoop

1. Issue with small files
2. Slow processing speed
3. Latency
4. No real time data processing
5. Support for batch processing only.
6. No caching
7. Not ease of use.
8. No delta iteration.

Amey.

NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 12

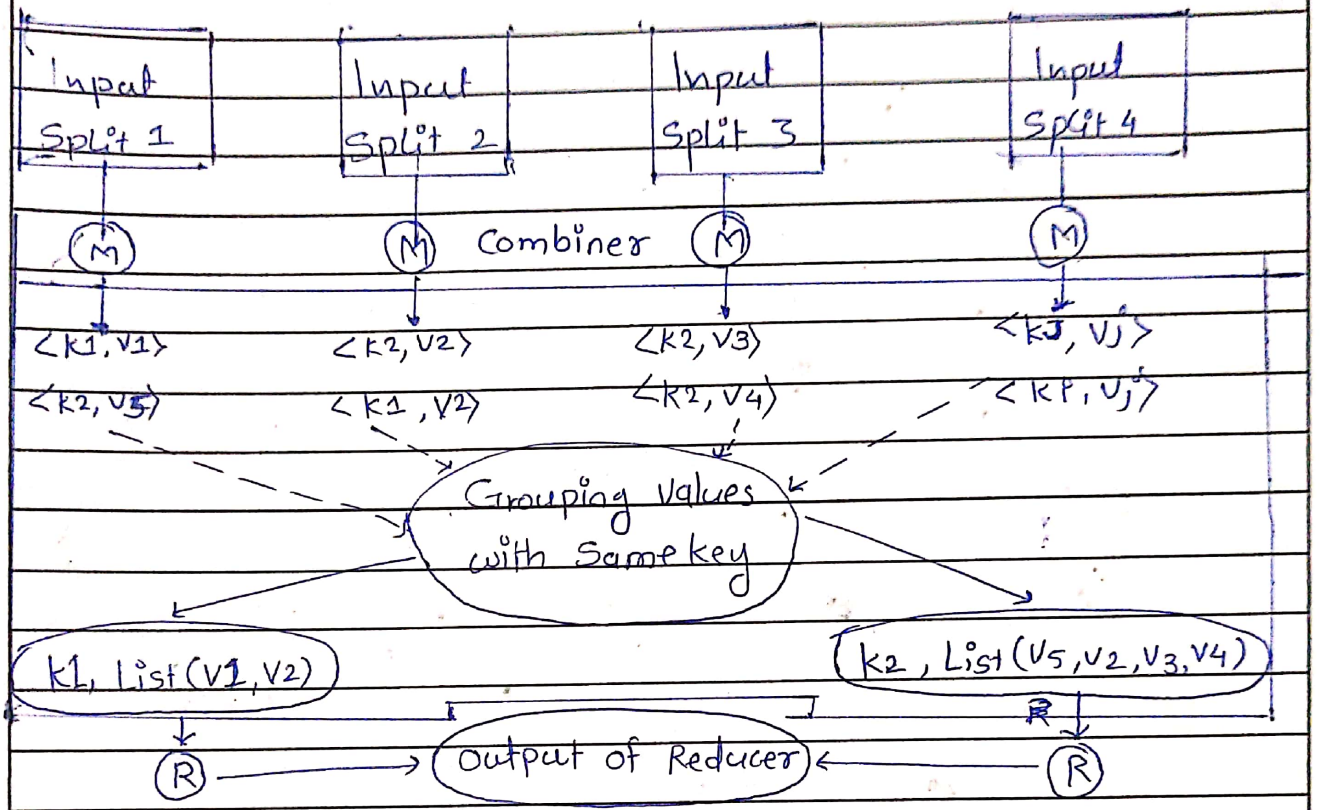
Q3 A] (ii)

Combiners

- A combiner is also known as semi-reducer.
- It is one of mediator between the mapper phase and the reducer phase.
- The use of combiners is totally optional.
- It accepts the output of map phase as an input and pass the key-value pair to the reduce operation.
- The main function of a Combiner is to summarize the map output records with the same key.
- It is also known as grouping by key
- The output (key-value collection) of the combiner will be sent over the network to the actual Reducer task as input.
- The Combiner class is used in between the Map class and the Reducer Class to reduce the volume of data transfer between Map and Reduce.
- Usually the output of the map task is large and the data transferred to the reduce task is high.

Amey.

- The following figure shows position and working mechanism of combiner.



Working

- A combiner does not have a predefined interface and it must implement the Reducer interface's reduce() method.
- A combiner operates on each map output key.
- It must have the same output key-value types as the Reducer class.
- A combiner can produce summary information from a large dataset because it replaces the original Map output.

Amey

Q3 B] (i)

MapReduce

- MapReduce is a software framework.
- MapReduce is the data processing layer of Hadoop.
- Similar to HDFS, MapReduce also exploits master/slave architecture in which JobTracker runs on master node and TaskTracker runs on each slave node.
- TaskTrackers are processes running on data nodes.
- These monitors the maps and reduce tasks executed on the node and coordinates with JobTracker.
- JobTracker monitors the entire MR job execution.
- JobTracker and TaskTracker are two essential process involved in MapReduce execution in MRv1 or Hadoop version 1.
- Both processes are now deprecated in MRv2 or Hadoop version 2 and replaced by resource manager, application master and node manager deamons.

Amey.

Example:

(A) File 1 : "Hello Anna Hello Elsa"

(B) File 2 : "Goodnight Anna Goodnight Elsa"

Operations:

(1) Map :

Map 1	Map 2
<Hello, 1>	<Goodnight, 1>
<Anna, 1>	<Anna, 1>
<Hello, 1>	<Goodnight, 1>
<Elsa, 1>	<Elsa, 1>

(2) Combine

Combine map 1	Combine Map 2
<Anna, 1>	<Anna, 1>
<Elsa, 1>	<Elsa, 1>
<Hello, 2>	<Goodnight, 2>

(3) Reduce

<Anna, 2>

<Elsa, 2>

<Goodnight, 2>

<Hello, 2>

Amey

NAME: AMEY MAHENDRA THAKUR | EXAM: SEMESTER - VII | SEAT NO.: 7278000

SUBJECT: BDA | DATE: 29-11-2021 | PAGE NO.: 16

Q 3 B] (ii)

Dead ends are Handled in Page Rank

- A dead end is a web page with no links out. The presence of dead ends will cause the page rank of some or all the pages to go to 0 in the iterative computation, including pages that are not dead ends.
- Dead ends can be eliminated before undertaking a page rank calculation by recursively dropping nodes with no arcs out. Note that dropping one node can cause another which linked only to it to become a dead end, so the process must be recursive
- Two approaches to deal with dead ends:

(1) We can drop the dead ends from the graph and also drop their incoming arcs. Doing so may create more dead ends which also have to be dropped recursively. However eventually we wind up with a strongly connected component (SCC) none of whose nodes are dead ends. Recursive deletion of dead ends will remove parts of the out component, tendrils and tubes but leave the SCC and the in-component as well as parts of any small isolated components.

Amey

NAME: AMEY MAHENDRA THAKUR | EXAM: SEMESTER - VII | SEAT NO.: 7278000

SUBJECT: BDA | DATE: 29-11-2021 | PAGE NO.: 17

(2) We can modify the process by which random surfers are assumed to move about the web. This method which we refer to as taxation also solves the problem of spider traps. Here, we modify the calculation of page rank by allowing each random surfer a small probability of teleporting to a random page rather than following an out link from their current page. This iterative step where we compute a new vector estimate of page rank v from the current page rank estimate v and the transition matrix M is

$$v' = \beta M v + \frac{(1-\beta)e}{n}$$

→ where β is the chosen constant - usually in the range of 0.8 to 0.9

→ e is a vector of all 1's with the appropriate number of components.

→ n is the number of nodes in the web graph

- The term $(1-\beta)e/n$ does not depend on the sum of the components of the vector v, there will always be some fraction of a surfer operating on the web. That is, when there are dead ends, the sum of the components of v may be less than 1, but it will never reach 0.

Amey

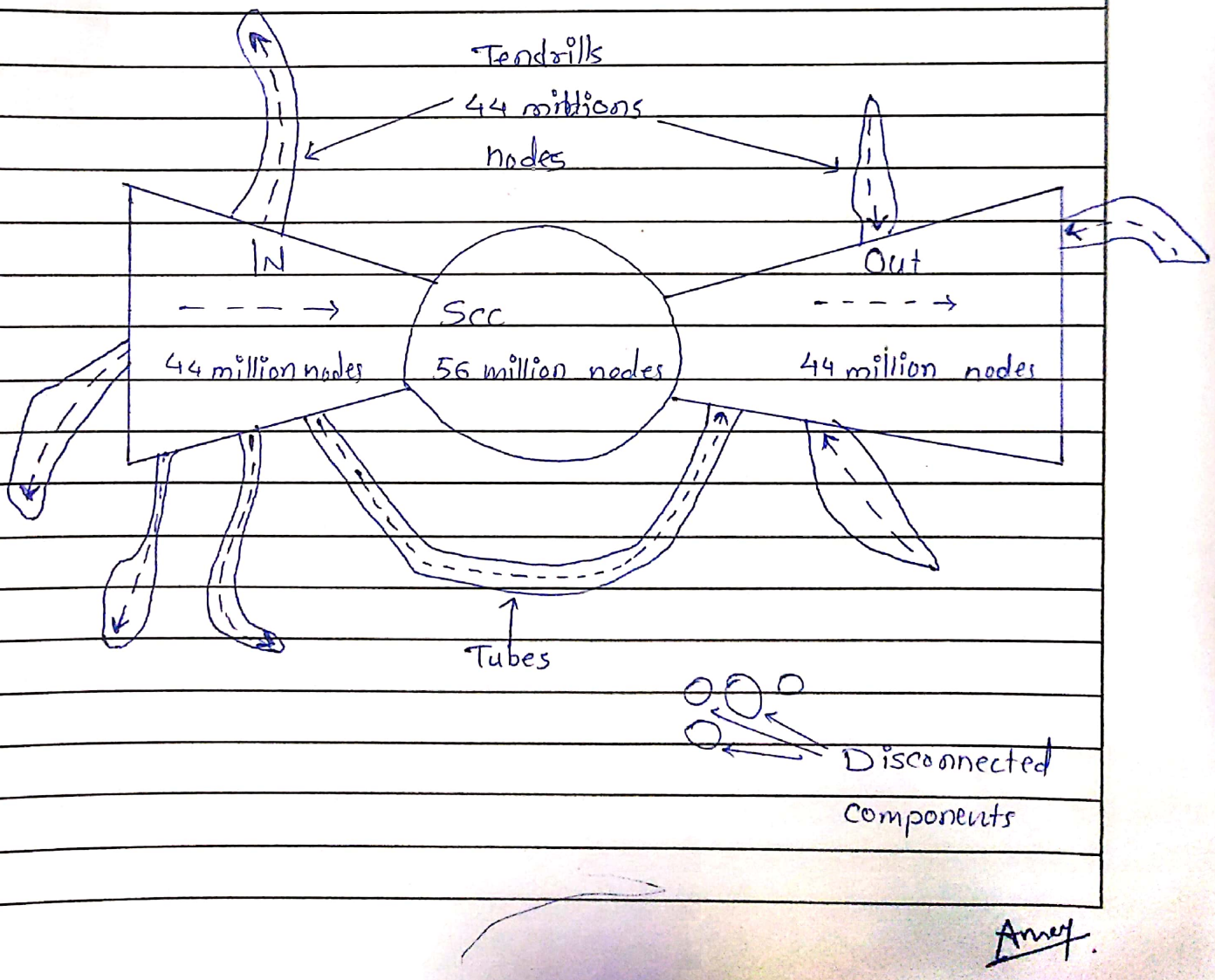
NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 18

Q4 A) (i)

Components of the web's Bow-tie structure

- If we consider web pages as vertices and hyperlinks as edges
- Then, the web can be represented as a directed graph.
- The structure below shows a large strongly connected component (SCC), but there were several other portions that were almost as large.
- Bow-tie structure of the web



Amey.

- The two regions of approximately equal size on the two sides of CORE are named as
  1. IN: Nodes that can reach the giant SCC but cannot be reached from it.
  2. Out: Nodes that can be reached from the giant SCC but cannot reach it.
- This structure of web is known as Bow-tie structure.
- There are pages that belong to none of IN, OUT or giant SCC. ie they can neither reach the giant SCC nor be reached from it.
- They are classified as.
  1. Tendrils
  2. Disconnected.

Amey.

NAME: AMEY MAHENDRA THAKUR | EXAM: SEMESTER - VII | SEAT NO.: 7278000

SUBJECT: BDA | DATE: 29-11-2021 | PAGE NO.: 20

Q4 A] (ii)

-

Flajolet - Martin Algorithm

- The problem of counting the distinct element can be solved with the help of ordinary hashing technique.
- A hash function will be applied to a given set which generate a bit - string as a result of hashing.
- A constraint should be applied to above process is, there should enough hashing results than elements present inside the set.
- The general procedure while applying hash function is to pick different hash function for and apply there every element in given data stream.
- The significant property of flash function is that, whenever applied to the same data element in a given data stream it will generate the same hash value.
- So, Flajolet - Martin algorithm has extended this hashing idea and properties to count distinct elements.
- The algorithm was introduced by Philippe Flajolet and G. Nigel Martin in their 1984.
- Flajolet Martin algorithm approximates the number of unique objects in a stream or a database in one pass.
- If the stream contains n elements with m of them unique, this algorithm runs in $O(n)$ time and needs $O(\log(m))$ memory.
- So the real innovation here is the memory usage, in that an exact brute-force algorithm would need $O(m)$ memory.
- As noted, this is an approximate algorithm.

Amey.

Algorithm

① Create a bit vector of sufficient length L, such that $2^L > n$, the number of elements in the stream. Usually a 64-bit vector is sufficient since 2^{64} is quite large for most purposes.

② The i^{th} bit in this vector/array represents whether we have seen a hash function value whose binary representation ends in 0^i. So initialize each bit to 0

③ Generate a good, random hash function that maps input to natural numbers.

④ Read input. For each word, hash it and determine the number of trailing zeros. If the number of trailing zeros is k, set the k^{th} bit in the bit vector to 1.

⑤ Once input is exhausted, get the index of the first 0 in the bit array. By the way, this is just the number of consecutive 1s. plus 1.

⑥ Calculate the number of unique words as $2^R/\phi$, where ϕ is 0.77351.

⑦ Standard deviation of R is a constant: $\sigma(R) = 1.12$.

⑧ This implies that our count can be off by a factor of 2 for 32% of the observations, off by a factor of 4 for 5% of the observations, off by a factor of 8 for 0.3% of the observations and so on.

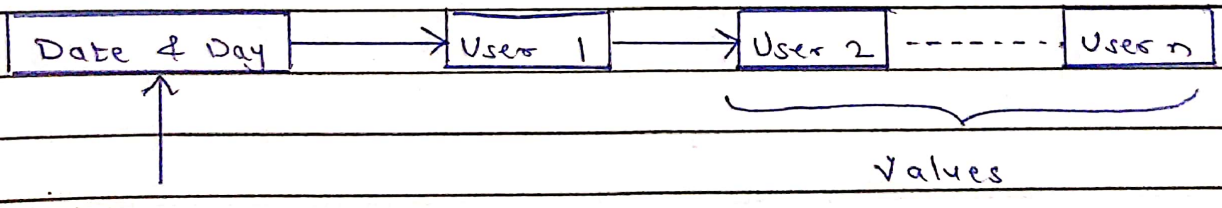
Amey

NAME: AMEY MAHENDRA THAKUR EXAM: SEMESTER - VII SEAT NO.: 7278000

SUBJECT: BDA DATE: 29-11-2021 PAGE NO.: 22

Count Distinct Problem

- In computer science, the count-distinct problem is the problem of finding the number of distinct elements in a data stream with repeated elements.
- Consider an example of counting number of newly added users who will view a given web page 1st time in the last month.
- Clearly, the standard approach to solve this problem is, make a list of all elements of user who has seen before in a given data stream.
- Convert the data elements in the list into some efficient data structure, more specifically a search structure such as tree.
- The major benefits of this kind of structure is we can have bucket like structure where element from some category will be found in some bucket.
- Eg. list of user who has seen a given web page on some data



- The addition of new user to an existing list according to decided key is very easy with this structure.
- So, it seems to be very easy to count the distinct element within the given stream.

Amey

- The major problem arises, when the number of distinct element are too much in number and to make it worst.
- If we have number of such data stream who will enter into the system at the same time.
- Eg. If organization like Yahoo or Google requires a count of user who will see their different web pages 1st time for a given month.
- Here, for every page we need to maintain above said problems which seems to be a complicated problem.
- To have alternative solution to this problem, we can have "scale out of machines".
- In this approach, we can add new commodity hardware to existing hardware so that load of processing and counting the distinct user on every web page of an organization will be distributed.
- Another additional thing is that we can have the use of secondary memory can batch systems.

Amey.

Q4. A] (iii)

Big data applications based on NoSQL

① Key Value Store Databases

- It is one of the most basic types of NoSQL databases.
- This kind of NoSQL database is used as a collection, dictionaries, associative arrays, etc.
- Data is stored in key/value pairs.
- It is designed in such a way to handle lots of data and heavy load.
- Key-value pair storage databases store data as a hash table where each key is unique and the value can be a JSON, BLOB (Binary Large Objects), string, etc.
- They work best for shopping cart contents.
- Examples:
  ① Azure Table storage (ATS)
  ② DynamoDB
- Limitations:
  ① It may work well for complex queries attempting to connect multiple relations of data.
  ② If data contains lot of many-to-many relationships, a key value store is likely to show poor performance.

Key : 1	ID : 420	First Name : Alex	
Key : 2	Email : alex123@gmail.com	Location : Mumbai	Age : 20
Key : 3	Facebook ID : Alex	Password : 12345678	Name : Alex

Example of unstructured data for user records

Amey.

② Column Store Database

- Instead of storing data in relational tuples, it is stored in cells grouped in columns.
- Column-oriented databases work on columns and are based on BigTable paper by Google.
- Every column is treated separately.
- Values of single column databases are stored contiguously.
- They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN, etc. as the data is readily available in a column.
- Column-based NoSQL databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs.
- Examples:
  ① HBase ② Cassandra ③ Hyper Table

EmpNO	DeptID	Hire Date	EmpName
1	1	2001-01-01	Alex
2	1	2002-02-01	Ronaldo
3	1	2002-04-01	Anna
4	2	2003-01-01	Messi
5	2	2004-01-01	Elsa

Row Oriented Database

1	1	2001-01-01	Alex
2	1	2002-02-01	Ronaldo
3	1	2002-04-01	Anna

Column Oriented Database

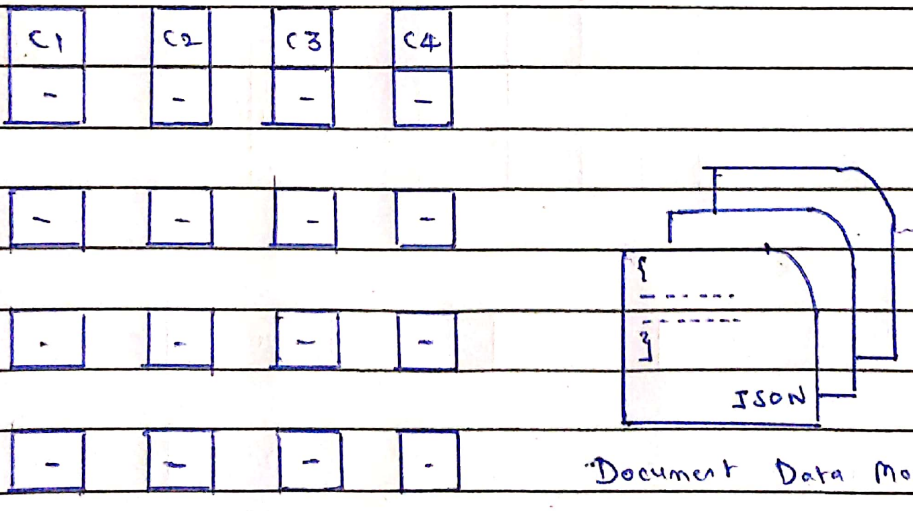
1	2	3	4	5
1	1	1	2	2
2001-01-01	2002-02-01	2002-04-01	2003-01-01	2004-01-01

Example of column store database

Amey

③ Document Database

- Document - oriented NoSQL DB stores and receives data as a key value pair but the value part is stored as a document.
- The document is stored in JSON or XML formats.
- Every document contains a unique key, used to retrieve the document.
- Key is used for storing, retrieving and managing document oriented information also known as semi-structured data
- Examples:
  1. MangoDB
  2. CouchDB.
- Limitations:
  1. It's challenging for document store to handle a transaction that on multiple documents.
  2. Document databases may not be good if data is required in aggregation.



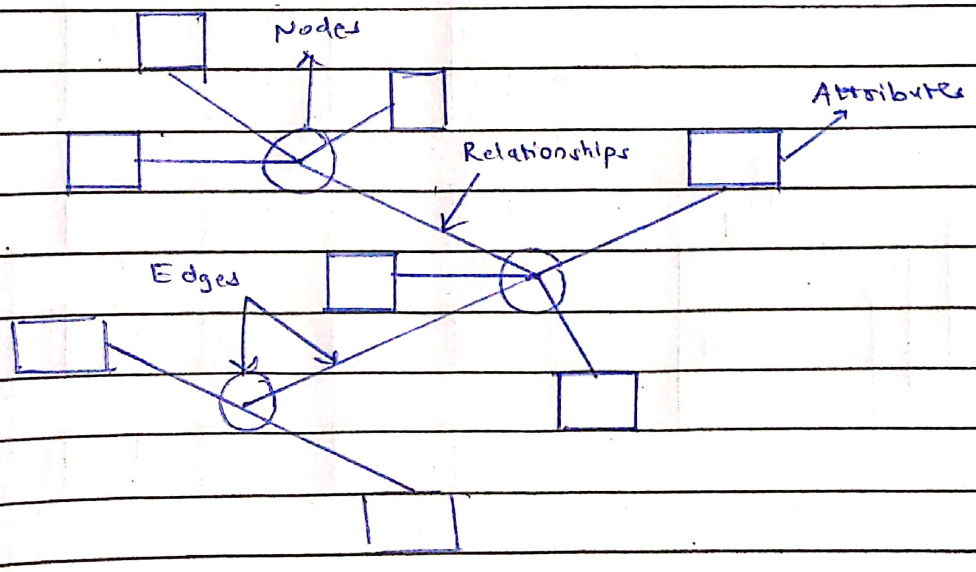
Relational Data Model Document Data Model

Example of Document Database

Amey

(4) Graph Database

- A graph database stores entities as well as the relations amongst those entities.
- The entity is stored as a node with the relationship as edges.
- An edge gives a relationship between nodes.
- Every node and edge has a unique identifier.
- Compared to a relational database where tables are loosely connected, a graph database is a multi-relational in nature.
- Traversing relationship is fast as they are already captured into the DB and there is no need to calculate them.
- Graph base database mostly used for social networks logistics and spatial data.
- Examples:
  1. Neo4j
  2. Infinite Graph
  3. OrientDB
  4. FlockDB



Example of Graph Database

Amey

Q4 B] (i)

DGIM

- DGIM stands for Datar - Gionis - Indyk - motwani Algorithm.
- It is designed to find the number 1's in a data set.
- This algorithm uses $O(\log^2 N)$ bits to represent a window of N bit, allows to estimate the number of 1's in the window with and error of no more than 50%.
- In DGIM algorithm, each bit that arrives has a timestamp, for the position at which it arrives.
- If the first bit has a timestamp 1, the second bit has a timestamp 2 and so on.
- The positions are recognized with the window size N. (The window sizes are usually taken as a multiple of 2)
- The windows are divided into buckets consisting of 1's and 0's.

Rules for forming the buckets:

1. The right side of the bucket should always start with 1. (If it starts with a 0, it is to be neglected.).
   Eg. 1001011 → a bucket of size 4, having four 1's and starting with 1 on it's right end.
2. Every bucket should have at least one 1, else no bucket can be formed.
3. All buckets should be in power of 2.
4. The buckets cannot decrease in size as we move to the left. (move in increasing order towards left)

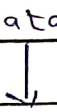
Amey

Q4 B] (ii)

(1) Big Data Clustering Algorithm

CURE Algorithm

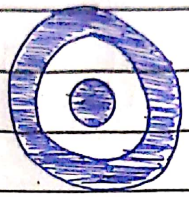
- CURE stands for Clustering Using Representatives Algorithm.
- It is an efficient data clustering algorithm for large databases.
- CURE Algorithm works better in spherical as well as non-spherical clusters.
- CURE uses random sampling and partitioning to speed up clustering.



CURE Overview

Amey

- The CURE algorithm is divided into phases.
  1. Initialization in CURE
  2. Completion of the CURE algorithm.

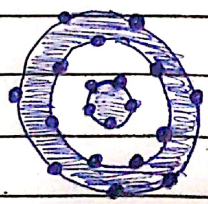


Two clusters, one surrounding the other.

- The inner cluster is an ordinary circle, while the second is the ring around the circle.
- This arrangement is not completely pathological.

Initialization in CURE

- Take a small sample of the data and cluster it in main memory.
- In principle, any clustering method could be used, but as CURE is designed to handle oddly shaped clusters, it is often advisable to use a hierarchical method in which clusters are merged when they have a close pair of points.
- Select a small set of points from each cluster to be representative points as shown in figure.

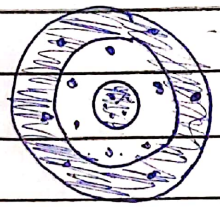


Select representative points from each cluster, as far from one another as possible.

Amey.

- These points should be chosen to be as far from one another as possible, using the k-means method.
- Move each of the representative points a fixed fraction of the distance between its location and the centroid of its cluster.
- Perhaps 20% is a good fraction to choose.

Note that this step requires a Euclidean space, since otherwise, there might not be any notion of a line between two points.



Moving the representative points 20% of the distance to the cluster's centroid.

Completion of the CURE Algorithm

- The next phase of CURE is to merge two clusters if they have a pair of representative points, one from each cluster, that are sufficiently close.
- The user may pick the distance that defines "close".
- This merging step can repeat, until there are no more sufficiently close clusters.

Amey

(2) Big Data Clustering Algorithm.

BDMO Algorithm / Stream Clustering Algorithm

- Stream clustering algorithm is also referred as BDMO algorithm.
- BDMO algorithm is designed by Babcock, Datar, Motwani and OCallaghan.
- It is based on K-means.
- The BDMO algorithm follows the concept of 'counting ones' method, which means that there is a window of length N on a binary stream and it counts the number of 1s that comes in the last K bits where $K \leq N$.
- The BDMO algorithm uses the bucket with allowable bucket sizes that forms a sequence where each size is twice of the previous size.
- In the algorithm, the number of points represents the size of the bucket.
- It does not consider that the sequence of allowable bucket sizes starts with 1 but consider only forming a sequence such as 2, 4, 6, 8, ... where each size is twice the previous size.
- For maintaining the buckets, the algorithm considers the size of the bucket with the power of two.
- In addition, the number of buckets of each size is either one or two that form a sequence of non-decreasing size.
- The buckets that are used in the algorithm, contains the size and timestamp of the most recent points of the stream.

Amey.

- Along with this, the bucket also contains a collection of records that represents the clusters into which the points of that bucket have been partitioned.
- This record contains the number of points in the cluster, the centroid, or clustroid of the cluster and other parameters that are required to merge and maintain the clusters.
- The major steps of the BDMO algorithm are as follows:
  1. Initializing buckets
  2. Merging buckets
  3. Answering queries.

Amey.