Total Marks of Question no.		Examiner 1	MSE 2 Solution

Space for Marks	Question No.	START WRITING HERE
		A.Y. 2024-25 (odd)
		Department : CMPN
		Semester : 7
		Subject : Big Data Analytics (BDA)
		Exam Date : 30/9/24
	1 A)	Columnar architecture style of Google Big Table : Columnar architecture style of Google BigTable allows it to:
		• Store data in sparse multidimensional table
		• Group related columns into column family, enabling efficient storage & retrieval
		• Support high throughput for read/write operations, especially for large scale applications
		• Effectively handle versions of data with timestamps.
		This architecture makes Big table ideal for scenarios like time-series data, web indexing, analytics & real-time monitoring.
		(Any 2 points above can be elaborated in detail)
	1 B)	Comparison of key value pair architecture of NoSQL & with Document oriented architecture of NoSQL.
		P.T.O.

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE

Aspect	Key-value	Document
1) Data model:	Simple key-value pair	Documents (JSON, XML, or) with fields & values
2) Query capabilities	Lookup by key only	Rich queries on fields, indexing & aggregation
3) Data structure	Unstructured (value can be anything)	Semi-structured, hierarchical, supports nested data
4) Flexibility	High flexibility with value formats but simple	High flexibility with nested & complex documents
5) Use cases	Session data, simple lookups etc.	Content management, Product catalogs and User profiles.

Ans c) DBMS architecture consist of following components:

1) Processor with standing Queues
2) Limited working storage
3) Archival storage
4) I/P for stream data
4) Interface for ad-hoc query.

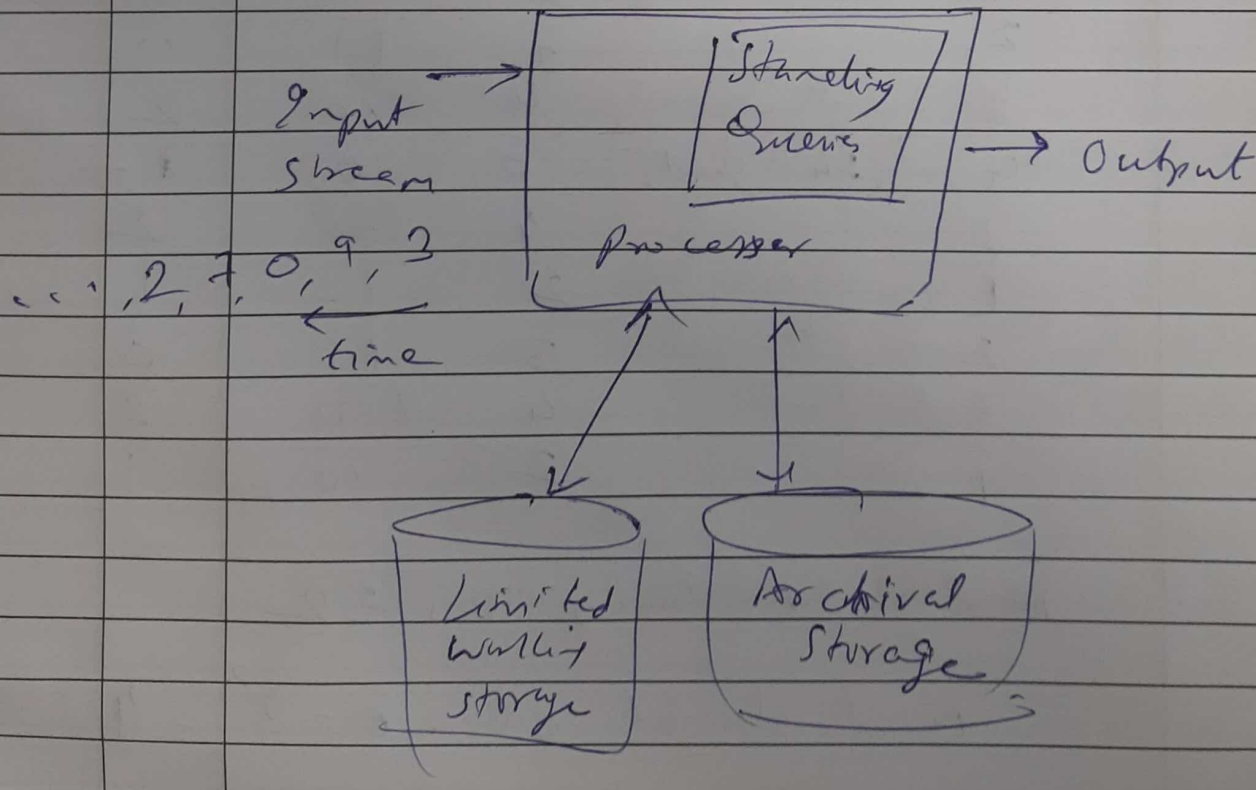
Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE

Data sources can be IoT devices, sensors and other data integrating entities. Input data can be in form of tuples like financial records etc..

Processor performs data cleaning, normalization & transformation to prepare data for processing.
Processed data required for standing queries are stored in limited working storage

Large storage is stored in Archival storage that can be used to address Adhoc queries.



Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE
		Standing queries are those that are asked about the data stream at all times.
		Example : Report each new maximum value ever seen in Stream S.
	Q2	
	(A)	No of distinct elements in a stream

$S = 1, 3, 2, 1, 2, 3, 4, 3, 1, 2, 3, 1$

$h(x) = (6x+1) \bmod 5$

Data	~~Binary~~ Hash	Binary	Trailing 0's
1	$(6+1) \bmod 5 = 2$	10	1
3	$(18+1) \bmod 5 = 4$	100	2
2	$(12+1) \bmod 5 = 3$	11	0
1	2	10	1
2	3	11	0
3	4	100	2
4	$(24+1) \bmod 5 = 0$	0	∞
3	4	100	2
1	2	10	1
2	3	11	0
3	4	100	2
1	2	10	1

Max no. of trailing 0's = 2

$\therefore$ No of distinct element $= 2^2 = 4$

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE
	2 B)	Bloom filter :-

Given a list of elements (say S), we need to determine whether an element x is in S or not.?

Solution using Bloom Filter

Consider $|S| = m$ & $|B| = n$
where B is hash table

Initially set all bits of B to 0s

For each element $y \in S$ use hash function h_i for $i = 1, \cdots, k$ and set $B[h_i(y)] = 1$. for each hash function

When an element x arrives
If $B[h_i(x)] == 1$ for all $i = 1, 2, \cdots K$
then declare that x is in S.
(i.e. if x hashes to all locations of B that are 1, then most likely it was seen earlier).
Otherwise declared x, as not in S.

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE

Example: Consider set of elements as
3, 5, 2, 6, 1

Hash functions

$h_1 = (3x + 1) \bmod 11$

$h_2 = (6x + 3) \bmod 11$

$h_3 = (5x + 2) \bmod 11$

Bucket size = 11

element 3 hashes to

$h_1(3) = 10 \bmod 11 = 10$

$h_2(3) = 21 \bmod 11 = 10$

$h_3(3) = 17 \bmod 11 = 6$

0	1	2	3	4	5	6	7	8	9	10
						1				1

Similarly other elements

	$h_1(k)$	$h_2(k)$	$h_3(k)$
5	5	0	5
2	7	4	1
1	4	9	7

Find bucket

0	1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	1	1	1	0	1	1

Consider element 4.

$h_1(4) = 12 \bmod 11 = 2$

∴ Bit 2 is 0 ⇒ 4 is not seen in set.

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE
	2 c)	The most suitable technique for the E-commerce website is to use DGIM algorithm.
		The DGIM (Data-Giomis-Indyk-Motwani) algorithm is an efficient method for tracking the number of 1's in the last N bits of a binary stream using limited memory.
		The E-ecommerce website can maintain a separate bit stream for each product. A transaction in which product is present is set to 1 and if absent then the bit is assumed as 0.
		For each '0' in i/p stream, DGIM does nothing. It creates a new window of size 1 only when the i/p stream has 1.
		Hence, considering the scenario of sparsity of data, DGIM would be suitable as it will create a new window of size 1 only for the set of products involved in the transaction.
		The algorithm also merges smaller buckets & hence has complexity $O(\log^2 N)$ to track & count 1's in last N bits.

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE
	3 A)	The most suitable filtering technique is Content based ~~filte~~ recommendation system. It can be used to suggest songs to customers by analyzing the characteristics or content of the ~~con~~ songs themselves, rather than relying ~~on~~ on other's preferences as in collaborative filtering. It focuses on matching songs with similar attributes to those the user has already liked.
		The first step for content based recommendation for Spotify would be to define and extract features. Each song is broken down into features such as genre of the song, singer/artist, music instrument used, theme of the song, melody, rhythm, tempo, language, etc...
		Step 2 is to create a user profile based on the songs that user has rated or heard. The profile contains weighted list of features from those songs.
		Eg. User 1 [Romantic, Bright, Sad, Kind, ...] having weights like [0.4, 0.8, 0.7, 0.9, ...].

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE

Step 3 is to compare new songs with user's profile i.e. finding songs having similar features. This can be achieved by using cosine or similarity.

Final step is to rank songs based on their similarity score & suggests songs with high similarity with user's profile.

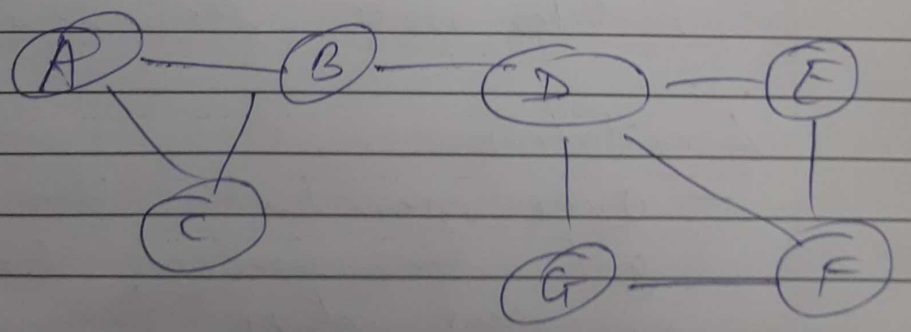
This technique will have advantage of personalization & can overcome the problem of cold start ie recommend the user song of a singer Arijit Singh which is having features like Romantic or Sad & in Hindi Language to users whose profile matches with the new realeased song.

Limitations.

- over specialization: User may only get recommendation based on their previous ratings thereby not giving exposure to user of other genre or artists.
- Feature Engineering: It relies on system's ability to extract meaningful features of the songs.

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE
	3B)	Most suitable technique for this social media platform to detect communities is to use Girvan-Newman (GN) algorithm.
		GN algorithm progressively removes edges that are most likely acting as a bridge between different communities.
		To determine the edge that connects 2 communities, GN algorithm computes edge betweenness. The edge that has highest edge betweenness score is considered to the bridge of 2 different communities and hence removed / discarded.
		Consider a social graph below:



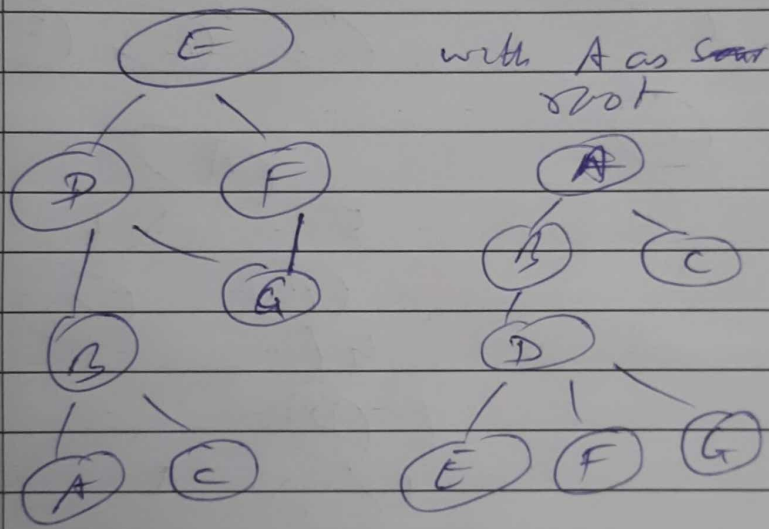
Step 1: Apply modified BFS. In this BFS, a node of level i+1 that is connected to more than

Total Marks of Question no.		Examiner 1	

Space for Marks	Question No.	START WRITING HERE

one node of level i, is considered to be child / reachable from both nodes.
Hence o/p of this modified BFS is not a steiner tree but a DAG.

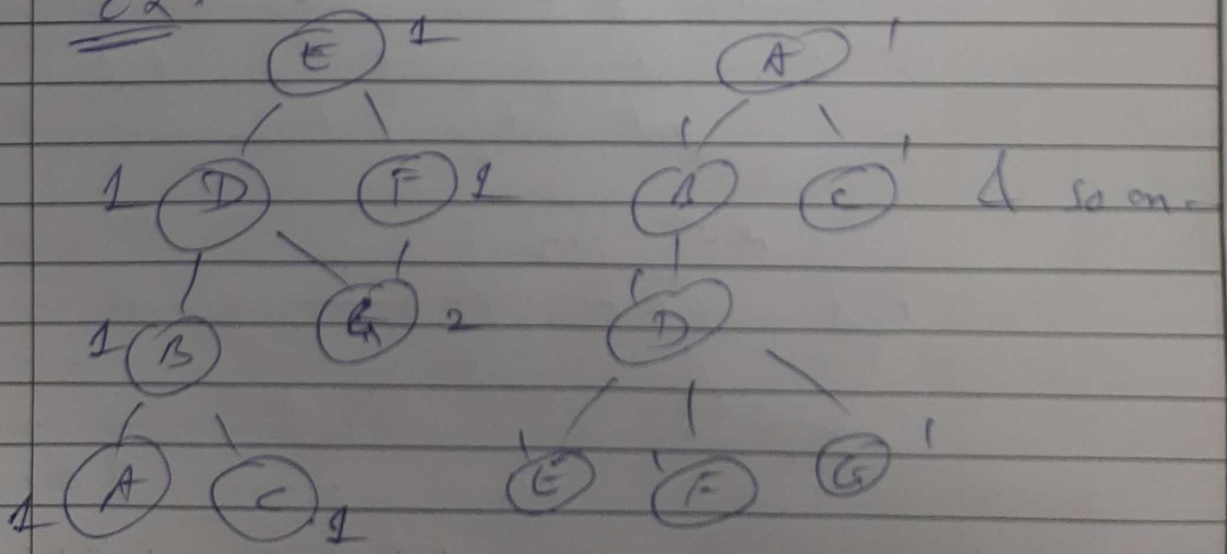
For the given example; with E as the root node, we get:



with A as root & so on - -

~~7th~~ Step 2: It labels the nodes with number of paths from the root.

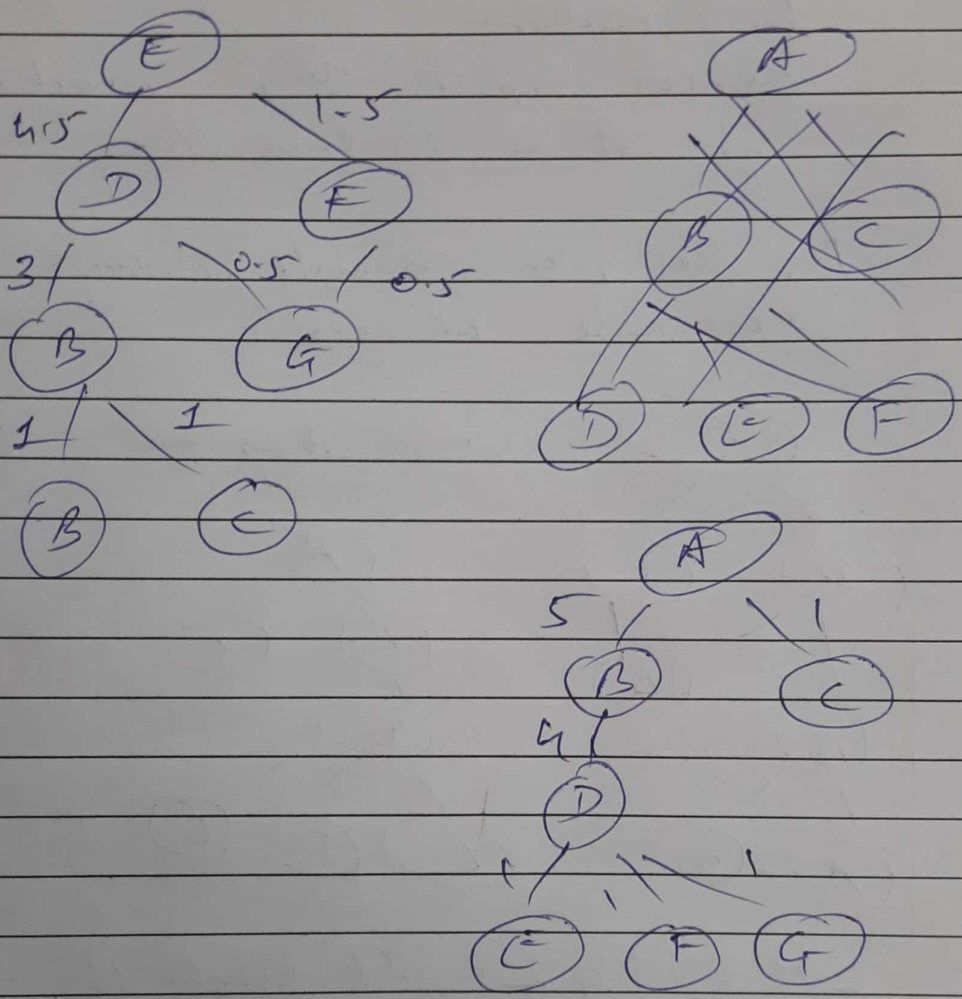
Eg:



Total Marks of Question no.		Examiner 1	

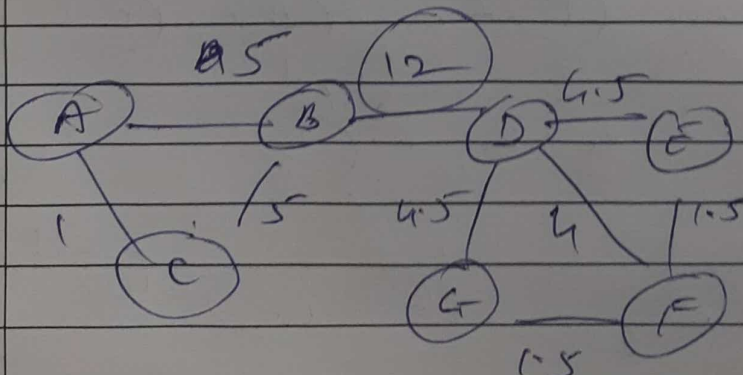
Space for Marks	Question No.	START WRITING HERE

Step 3 : It calculates credit from leaf to root. For example:



4 soon

The above steps after repeating for all roots, their edge contribution is summed up and divided by 2 to get edge between vertices. For example:



The edge BD is discarded to get 2 communities {A, B, C} & {D, E, F, G}.