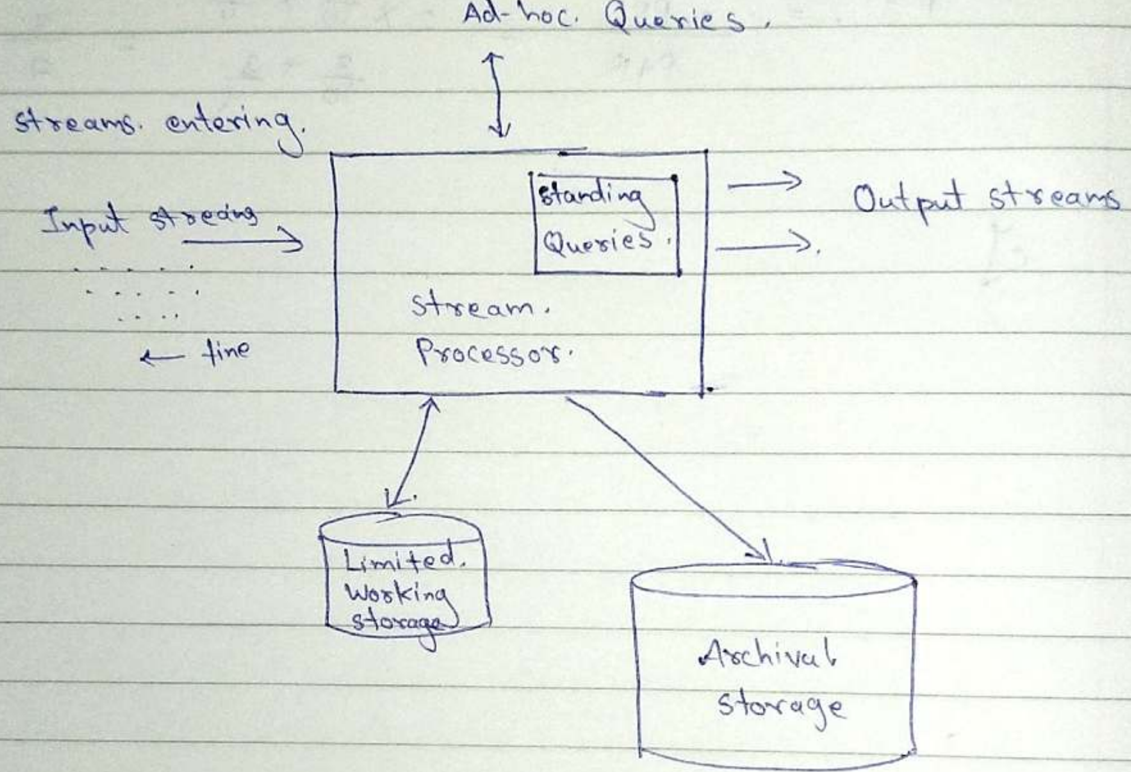
BDA notes 2022-2023

Big Data Analytics (University of Mumbai)

Q17. Explain with Block diagram Data Stream Management System.

Ans A data stream management system (DSMS) is a computer saftware system to manage continous data streams. It is similar to a database management system (DBMS), which is however, designed for static data in conventional databases.



Let us see components of the system

- Stream processor :- ~~The input streams.~~
  All types of processing such as sampling, cleaning, filtering, and querying on the input data are done. Two types of queries are supported. ie. standing queries & adhoc queries.

- standing queries: Is a query which is stared in a designated place inside the stream processar. The standing queries are excecuted whenever the conditions for the particular query become true.

- Ad-hoc queries:- An ad-hoc is not predefined and is issued on the go at the current state of the streams. The nature of the ad-hoc queries cannot be determined in advance.

For eg, social networking website like facebook may want to know the no of. unique active users over the past one month

Fig. 5.1.2 : A data-stream-management system

Let us understand the purpose of each of the components of the system :

1. **Input streams :** The input streams have the following characteristics:

   - There can be one or more number of input streams entering the system.
   - The streams can have different data types.
   - The rate of data flow of each stream may be different.
   - Within a stream the time interval between the arrival of data items may differ. For example, suppose the second data item arrives after 2 ms from the arrival of the first data item, then it is not necessary that the third data item will also arrive after 2 ms from the arrival of the second data item. It may arrive earlier or even later.

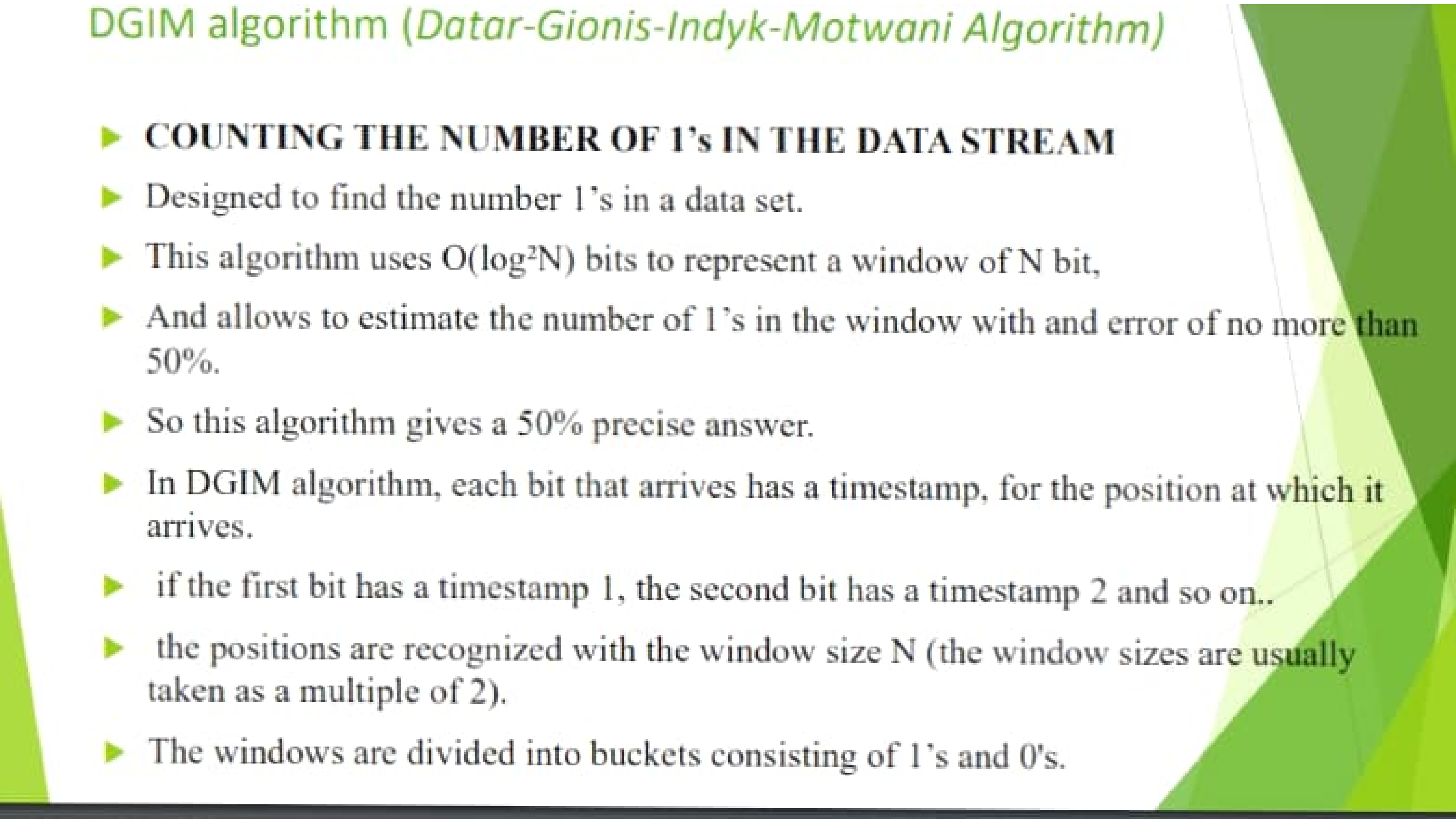
2. **Stream processor :** All types of processing such as sampling, cleaning, filtering, and querying on the input stream data are done here. Two types of queries are supported which are standing queries and ad-hoc queries. We shall discuss both the query types in details in the upcoming section.

3. **Working storage :** A limited memory such as a disk or main memory is used as the working storage for storing parts or summaries of streams so that queries can be executed. If faster processing is needed, main memory is used otherwise secondary storage disk is used. As the working storage is limited in size, it is not possible to store all the data received from all the streams.

4. **Archival storage :** The archival store is a large storage area in which the streams may be archived but execution of queries directly on the archival store is not supported. Also, the fetching of data from this store takes a lot of time as compared to the fetching of data from the working store.

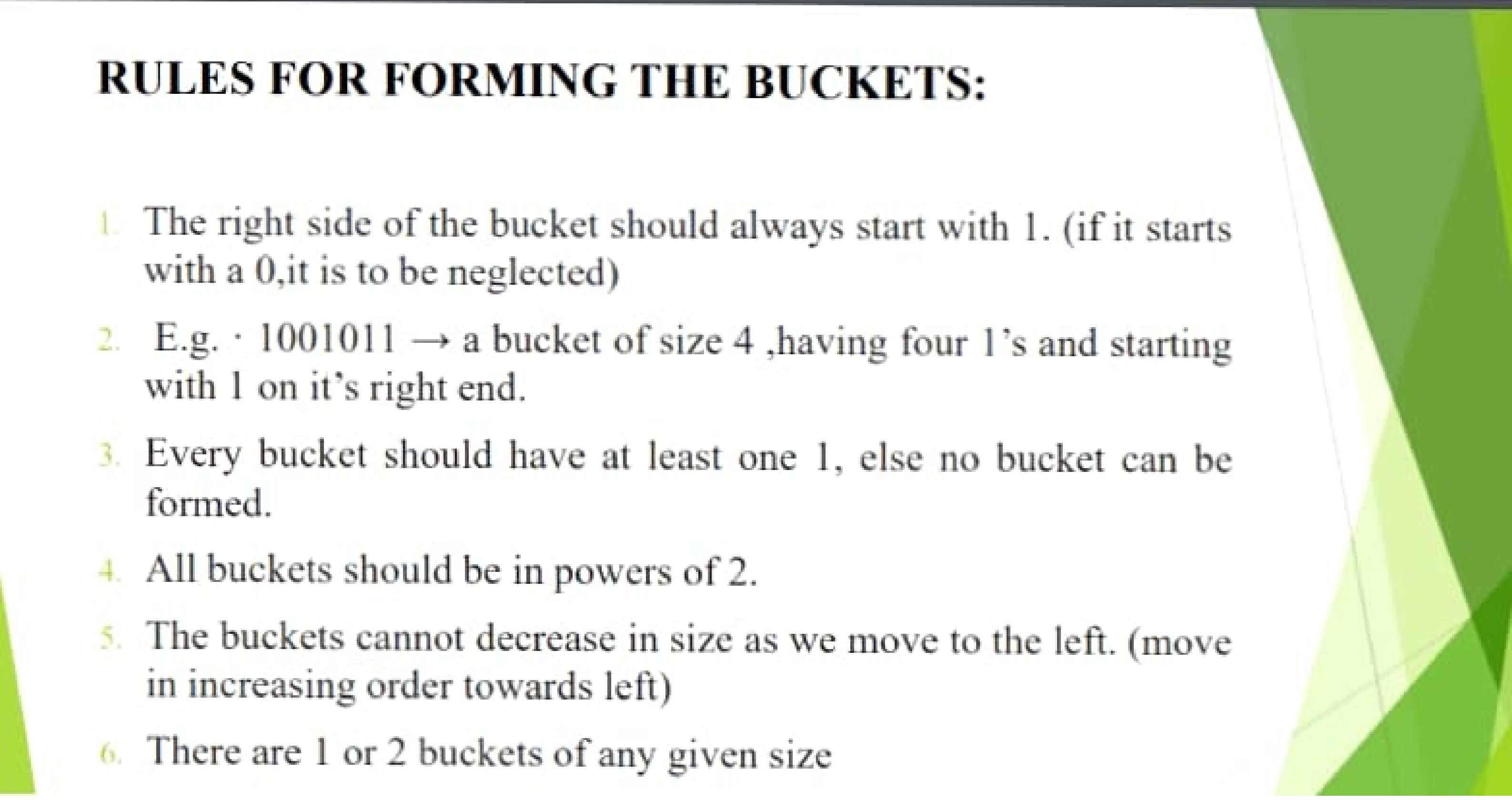
5. **Output streams :** The output consists of the fully processed streams and the results of the execution of queries on the streams.

- The difference between a conventional database-management system and a data-stream-management system is that in case of the database-management system all of the data is available on the disk and the system can control the rate of data reads. On the other hand, in case of the data-stream-management system the rate of arrival of data is not in the control of the system and the system has to take care of the possibilities of data getting lost and take the necessary precautionary measures.



DGIM algorithm (*Datar-Gionis-Indyk-Motwani Algorithm)*

- **COUNTING THE NUMBER OF 1's IN THE DATA STREAM**
- 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,
- And allows to estimate the number of 1's in the window with and error of no more than 50%.
- So this algorithm gives a 50% precise answer.
- 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.

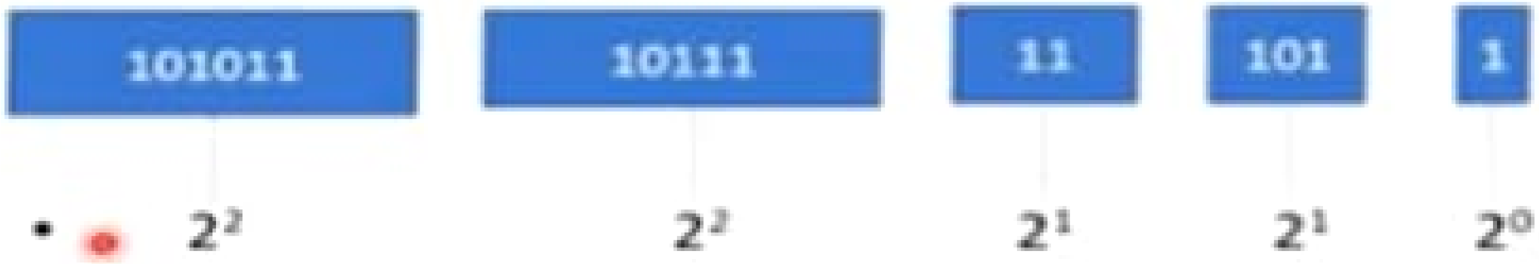


RULES FOR FORMING THE BUCKETS:

1. The right side of the bucket should always start with 1. (if it starts with a 0,it is to be neglected)
2. E.g. · 1001011 → a bucket of size 4 ,having four 1's and starting with 1 on it's right end.
3. Every bucket should have at least one 1, else no bucket can be formed.
4. All buckets should be in powers of 2.
5. The buckets cannot decrease in size as we move to the left. (move in increasing order towards left)
6. There are 1 or 2 buckets of any given size

Example 1:

- Input stream
- 101011000101110110010110
- N=24



- 2^2 2^2 2^1 2^1 2^0

- 101011 000 10111 0 11 00 101 1 0

Q3].

Ans

IP stream $x = \{1,3,2,1,2,3,4,3,1,2,3,1\}$.

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

Step 1]. Find h value for all values.

$h(1) = (6\times1+1) \bmod 5 = 2$

$h(3) = 4$

$h(2) = 2\ 3$

$h(1) = 2$

$h(2) = 3$

$h(3) = 4$

$h(4) = 0$.

$h(3) = 4$

$h(1) = 2$

$h(2) = 3$

$h(3) = 4$

$h(1) = 2$

Step 2 :- Converting the hash value into binary

$h(1) = 2 = 010$

$h(3) = 4. = 100.$

$h(2) = 3 = 011$

$h(1) = 2 = 010$

$h(2) = 3 = 011$

$h(3) = 4 = 100$

$h(4) = 0 = 0000$

$h(3) = 4 = 100$

$h(1) = 2 = 010$

$h(2) = 3 = 011$

$h(3) = 4 = 100.$

$h(1) = 2 = 010.$

Step 3: finding the no of trailing zeros

$h(1) = 1$

$h(3) = 2$

$h(2) = 0$

$h(1) = 1$

$h(2) = 0$

$h(3) = 2$

$h(4) = 0$

$h(3) = 2$

$h(1) = 1$

$h(2) = 0$

$h(3) = 2$

$h(1) = 1$

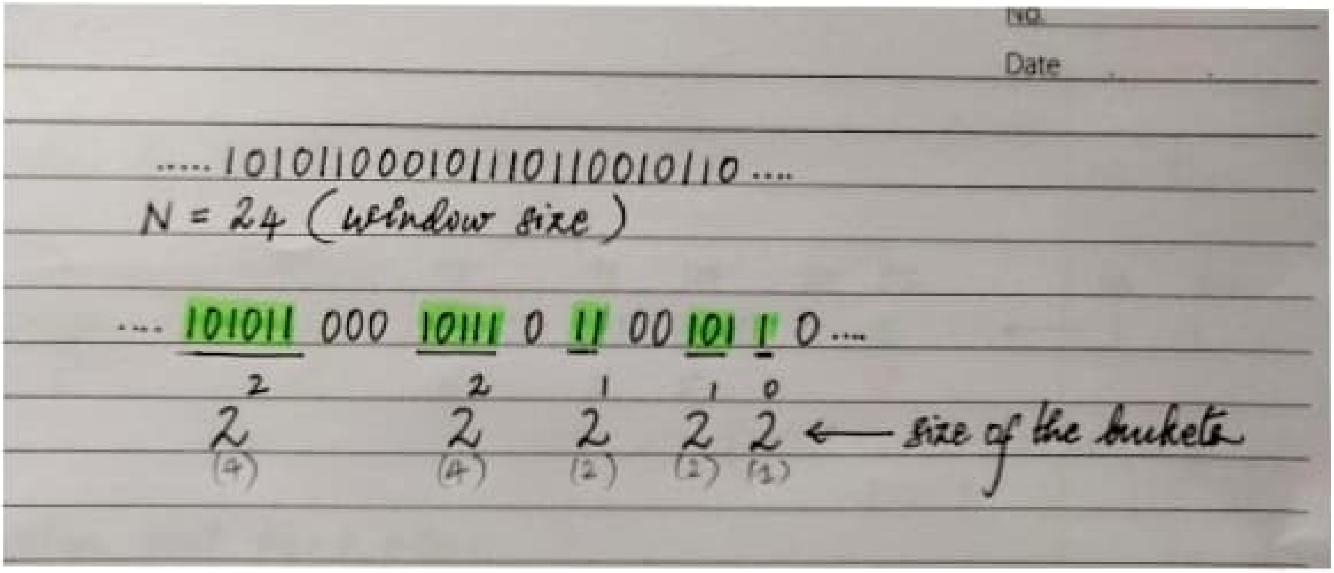
Step 5: Calculating the distinct element

r is the max no of trailing 0's

$$2^r = 2^2 = \underline{\underline{4}}$$

Let us take an example to understand the algorithm.

- Estimating the number of 1's and counting the buckets in the given data stream.



No.

Date

.....101011000101110110010110....

N = 24 (window size)

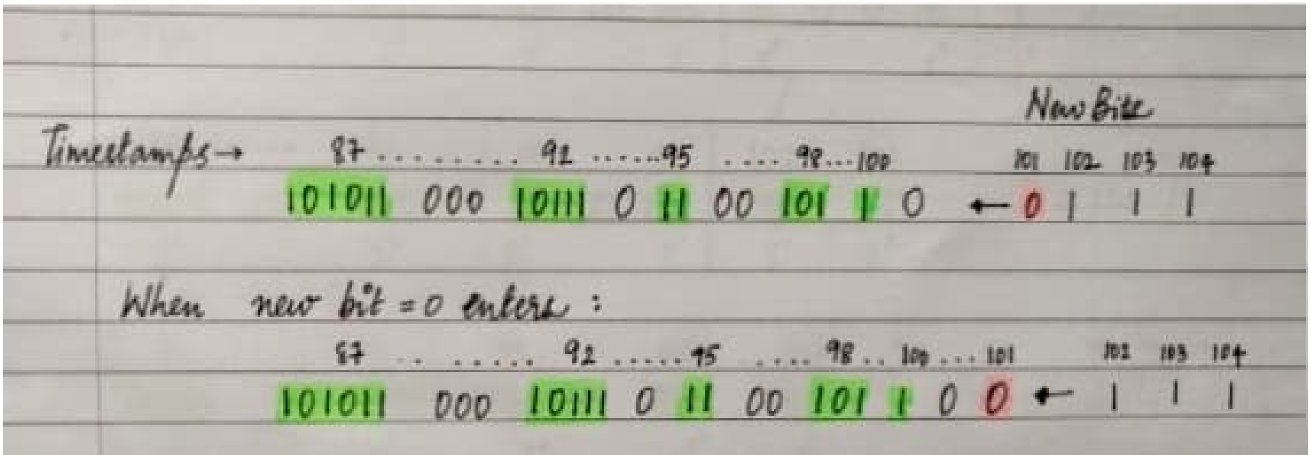
.... $\underline{101011}$ 000 $\underline{10111}$ 0 $\underline{11}$ 00 $\underline{101}$ $\underline{1}$ 0....

2^2 2^2 2^1 2^1 2^0 ← size of the buckets

(4) (4) (2) (2) (1)

This picture shows how we can form the buckets based on the number of ones by following the rules.

In the given data stream let us assume the new bit arrives from the right. When the new bit = 0



New Bits

Timestamps → 87.......... 92 ......95 ..... 98...100 101 102 103 104

101011 000 10111 0 11 00 101 1 0 ← 0 1 1 1

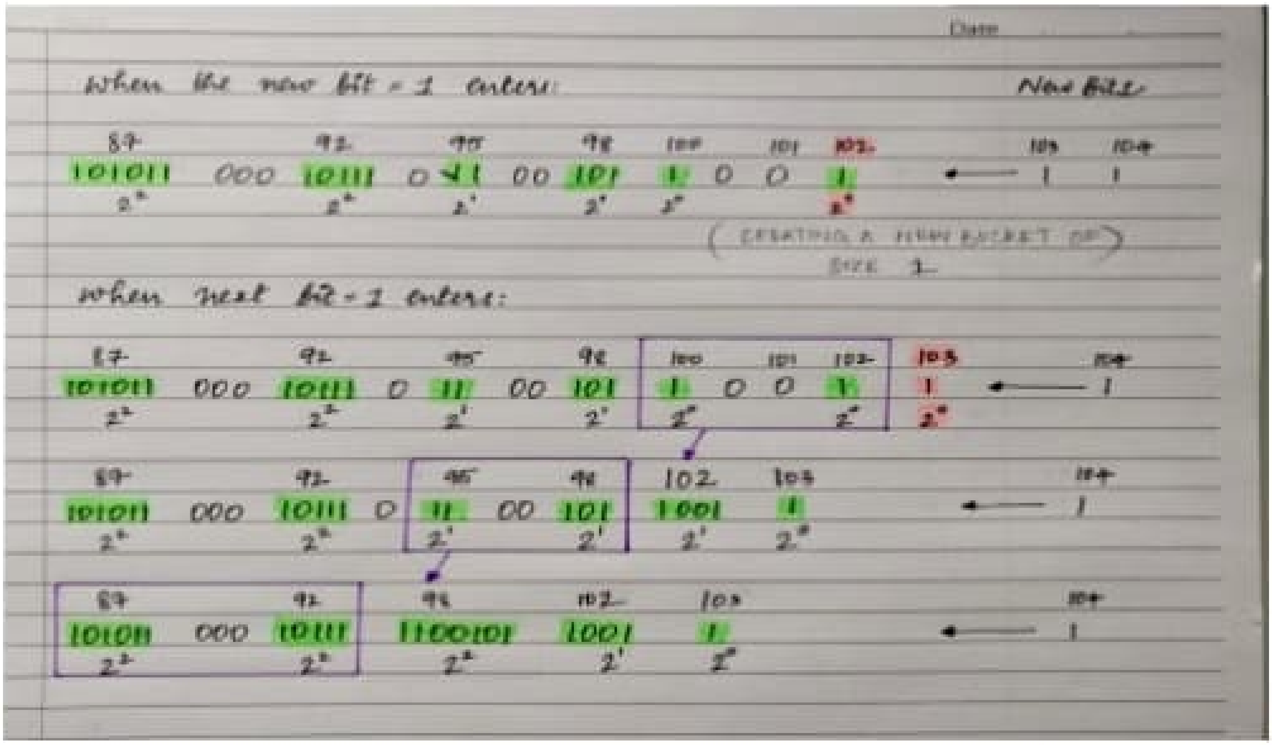
When new bit = 0 enters :

87 .. .. 92 ......95 ..... 98 .. 100 ... 101 102 103 104

101011 000 10111 0 11 00 101 1 0 0 ← 1 1 1

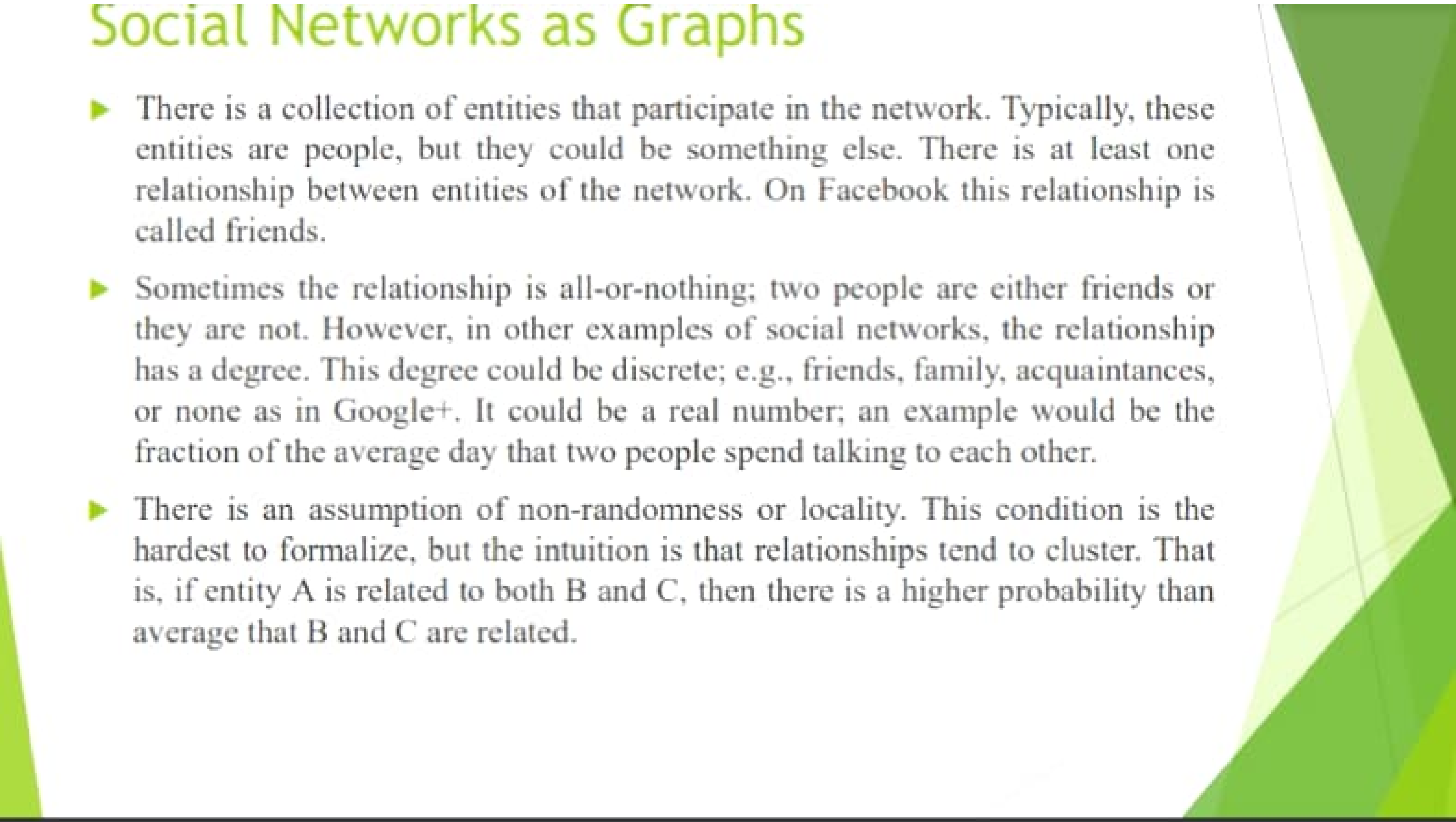
After the new bit (0) arrives with a time stamp 101, there is no change in the buckets.

But what if the new bit that arrives is 1, then we need to make changes.



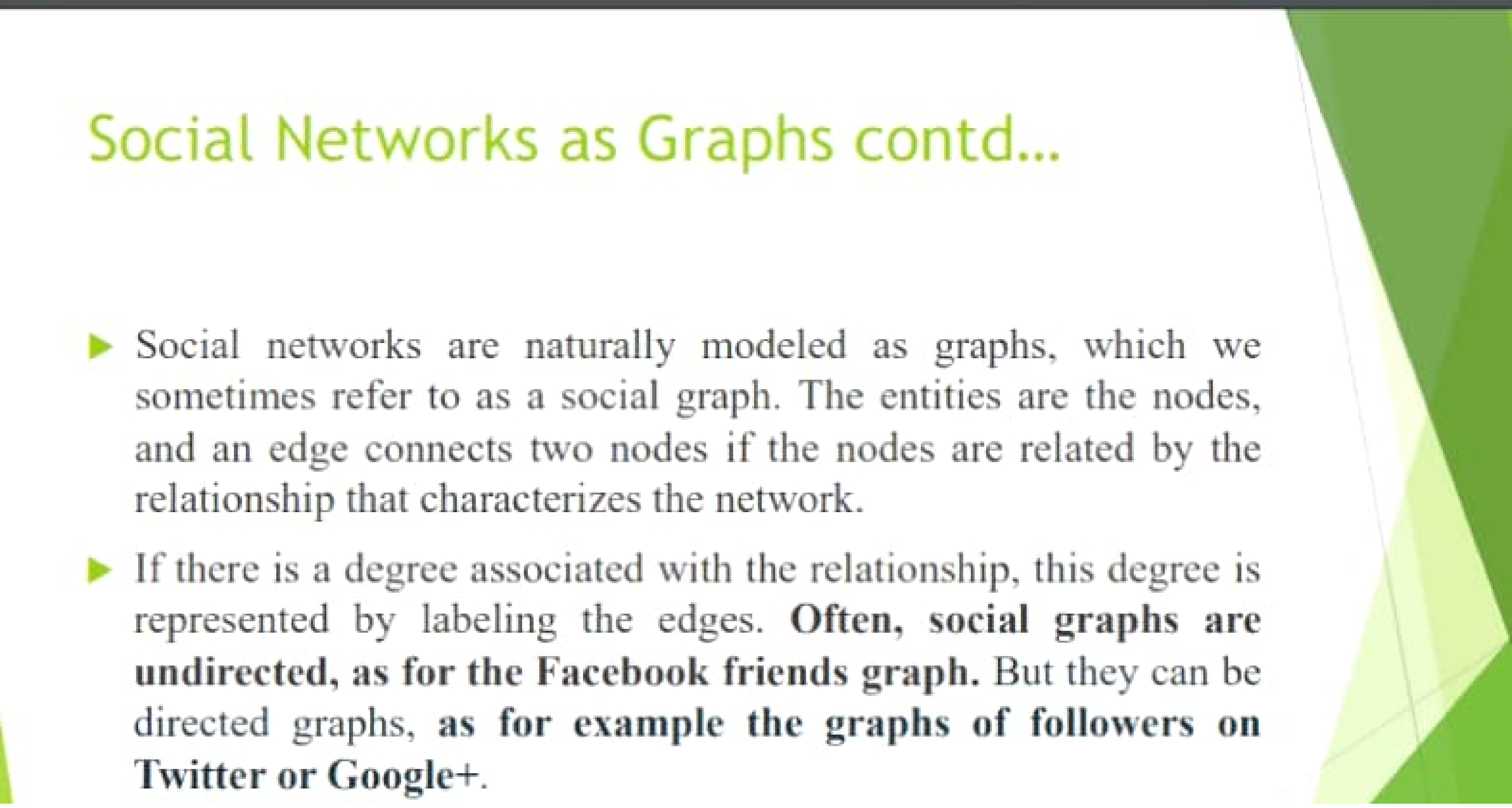
Create a new bucket with the current timestamp and size 1.

- If there was only one bucket of size 1, then nothing more needs to be done. However, if there are now three buckets of size 1(buckets with timestamp 100,102, 103 in the second step in the picture) We fix the problem by combining the leftmost(earliest) two buckets of size 1. (purple box)
- To combine any two adjacent buckets of the same size, replace them by one bucket of twice the size. The timestamp of the new bucket is the timestamp of the rightmost of the two buckets.
- Now, sometimes combining two buckets of size 1 may create a third bucket of size 2. If so, we combine the leftmost two buckets of size 2 into a bucket of size 4. This process may ripple through the bucket sizes.



Social Networks as Graphs

- There is a collection of entities that participate in the network. Typically, these entities are people, but they could be something else. There is at least one relationship between entities of the network. On Facebook this relationship is called friends.
- Sometimes the relationship is all-or-nothing; two people are either friends or they are not. However, in other examples of social networks, the relationship has a degree. This degree could be discrete; e.g., friends, family, acquaintances, or none as in Google+. It could be a real number; an example would be the fraction of the average day that two people spend talking to each other.
- There is an assumption of non-randomness or locality. This condition is the hardest to formalize, but the intuition is that relationships tend to cluster. That is, if entity A is related to both B and C, then there is a higher probability than average that B and C are related.



Social Networks as Graphs contd…

- Social networks are naturally modeled as graphs, which we sometimes refer to as a social graph. The entities are the nodes, and an edge connects two nodes if the nodes are related by the relationship that characterizes the network.
- If there is a degree associated with the relationship, this degree is represented by labeling the edges. **Often, social graphs are undirected, as for the Facebook friends graph.** But they can be directed graphs, **as for example the graphs of followers on Twitter or Google+**.

Discovery of communities deals with large number edges search from a graph.

Finding cliques

- Cliques can be defined as a set of nodes having edges between any two of vertices.
- To find clique is quite difficult task. To find largest set of vertices where any two vertices needs to be connected within a graph is known as maximum clique.

10.4.1 Bipirate Graph

- It is graph having vertices which can be partitioned into two disjoint sets suppose set V and set U. Both V and U sets are not necessary of having same size.
- A graph is said to be bipirate if and only if it does not posses a cycle of an odd length.

Example :

Suppose we have 5 engines and 5 mechanics where each mechanic has different skills and can handle different engine by vertices in U. An edge between two vertices to shows that the mechanics has necessary skill to operate the engine which it is linked. By determining maximum matching we can maximize the number of engines being operated by workforce.

10.4.2 Complete Bipirate Graph

- A graph $K_{m, n}$ is said to be complete bipirate graph as its vertex set partitioned into two subsets of m and n vertices respectively.
- Two vertices are connected if they belong to different subsets.

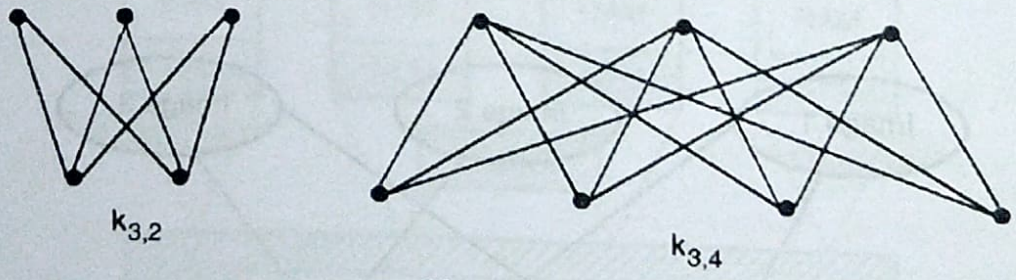


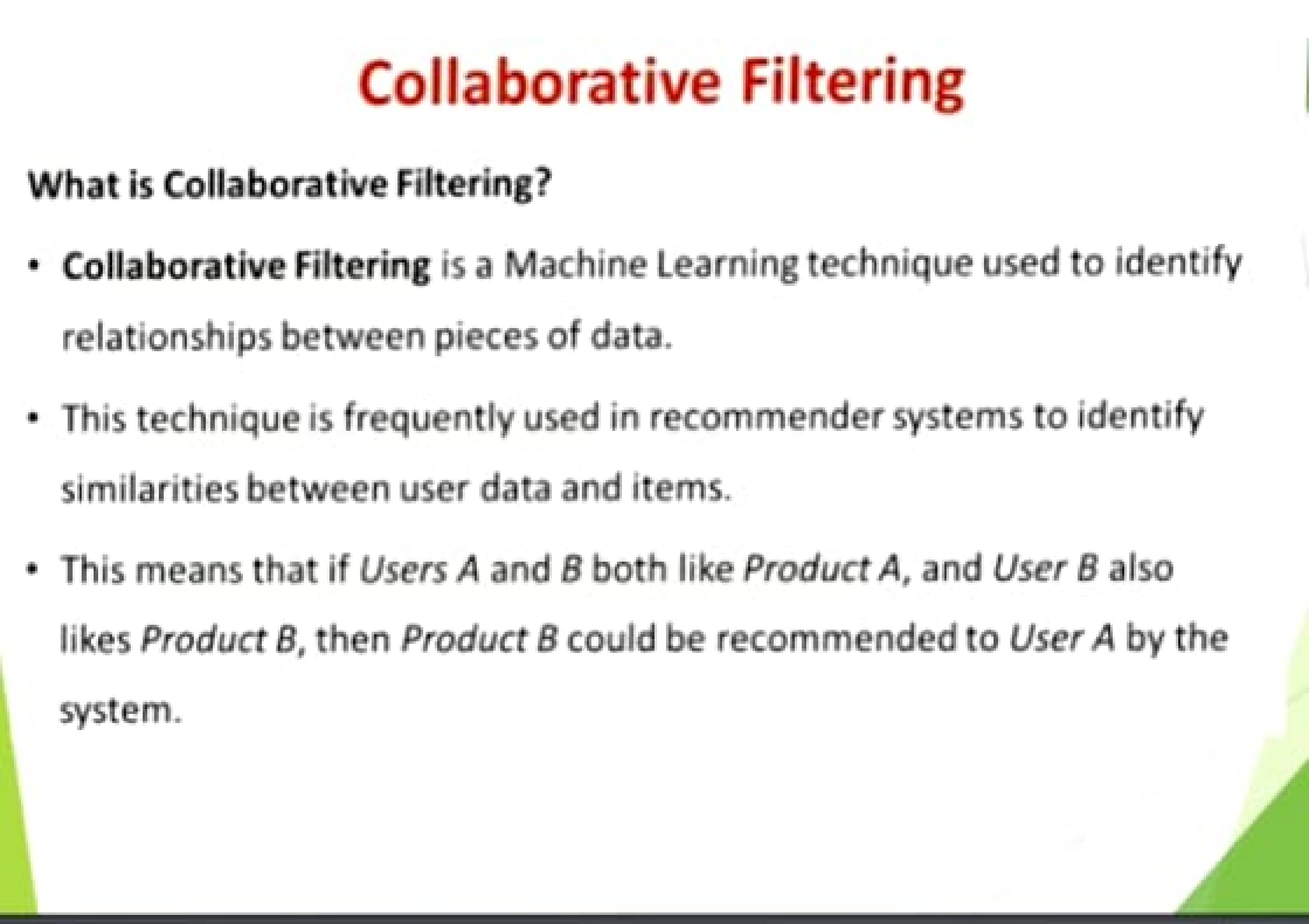
Fig. 10.4.1

Collaborative Filtering

- The recommendations are done based on the user's behavior. History of the user play an important role.
- The underlying assumption of the collaborative filtering approach is that if A and B buy similar products, A is more likely to buy a product that B has bought than a product which a random person has bought.
- Unlike content based, there are no features corresponding to users or items here. All we have is the **Utility Matrix**. This is what it looks like:

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

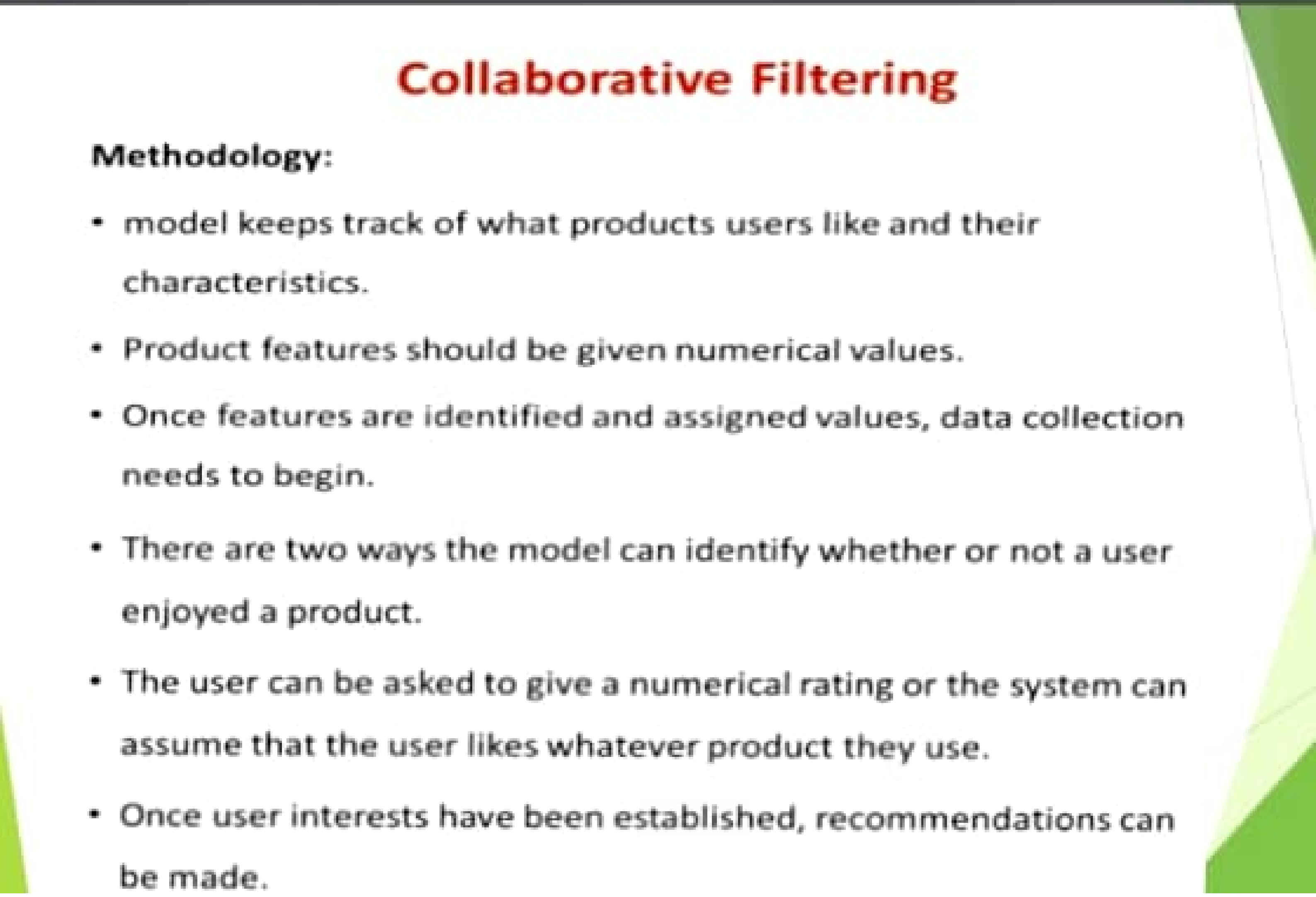
A, B, C, D are the **users**, and the **columns represent movies**. The values represent ratings (1–5) a user has given a movie. In other cases, these values could be 0/1 depending on whether the user watched the movie or not. There are a 2 broad categories that collaborative filtering can be split into:



Collaborative Filtering

What is Collaborative Filtering?

- **Collaborative Filtering** is a Machine Learning technique used to identify relationships between pieces of data.
- This technique is frequently used in recommender systems to identify similarities between user data and items.
- This means that if *Users A* and *B* both like *Product A*, and *User B* also likes *Product B*, then *Product B* could be recommended to *User A* by the system.



Collaborative Filtering

Methodology:

- model keeps track of what products users like and their characteristics.
- Product features should be given numerical values.
- Once features are identified and assigned values, data collection needs to begin.
- There are two ways the model can identify whether or not a user enjoyed a product.
- The user can be asked to give a numerical rating or the system can assume that the user likes whatever product they use.
- Once user interests have been established, recommendations can be made.

Q. Write any two pros and cons for Collaborative filtering and

9.4.1 Collaborative Filtering

Pros

(i) No knowledge engineering efforts needed.

(ii) Serendipity in results.

(iii) Continuous learning for market process.

Cons

(i) Rating feedback is required.

(ii) New items and users faces to cold start.

Q.7] Types of data visualization in R.

Ans Data visualization is a technique used to deliver insights in data using visual cues such as graphs, charts, maps and many others. ~~This~~

- This is useful as it helps in ~~creating~~ intuitive and easy understanding of the large quantities of ~~da~~ data & there by make better decisions.
- In simple terms representing data into statistical format such as graph, charts, etc.
- There are various types of data visualization :-

~~1] Line graph;~~

1] Bar plot:- There are 2 types of bar plots ie horizontal & vertical which represent data points as horizontal and vertical bars of certain lengths respect to the value of data item. By setting the horiz ~~or~~ parameters to true or false, we can get graph accordingly.

It is used to ~~do~~ do a comparative study.

2] Histogram:-

A histogram is like bar chart as it uses bars of varying height to represent data distribution. In histograms, values are grouped into consecutive intervals called bins. Continous values are grouped and displayed ~~wh~~ in bins whose size can be varied.

3] Box plot:-

The statistical summary of the given data is presented using this type of graph. It depicts information like minimum, maximum, median, etc:

- It is used to give a comprehensive statistical description of the data through visual cue.

4] **Scatterplot** :- It is composed of many points on a Cartesian plane. Each point denotes the value taken by 2 parameters and helps us easily identify the relationship between them.

5] **3D graphs** :-

Can also ~~graphs~~ graphs to create 3D perspective view. , we can achieve this prep () function.

~~This~~. It projects the 3D coordinate into the 2D plane using homogenous 4D coordinates.

Page No.
Date

Q9) Feature of R programming language

1) R programming is used as a leading tool for Machine learning, Statistics and data Analysis.

1) Features of R:

1) Basic Statistics:
The most Common basic Statistics terms are the mean mode and median. These are all known as "Measures of Central Tendency". So using R language we Can measure Central Tendency very easily.

2) Static graphics:
R is rich with facilities for Creating and developing interesting Static graph. R Contains functionality for many plot types including graphic maps, biplots, and mosaic plots

3) Probability distribution:
Probability distribution play a vital role in Statistics and by using R we Can easily handle Various type of probability distribution Such Binomial Distribution, Normal Distribution.

4) Data Analysis:-
It provide a large, coherent and integrated Collection of tools for data Analysis

5) R-Packages :-

one of th major feature of R is it has a wide Availability of librariss. R ha CRAN (Cmprehensive R Archive Network) which is a repository holding more than 10,0000 packages

6) Distributed Computing :-

Distributed Computing is a model in which Components of a Software System are shared among multiple Computers to improve efficiency and performance. Two New package ddR and multiplyr.

Q8) Built in function in R ?

→ 1) The function which are already Created or defined in the programming framework are known as a built in function.

2) The built in function are devided into the following Categories based on their functionality.

3) 1) Math functions

2) String functions

3) ~~Character fuction~~

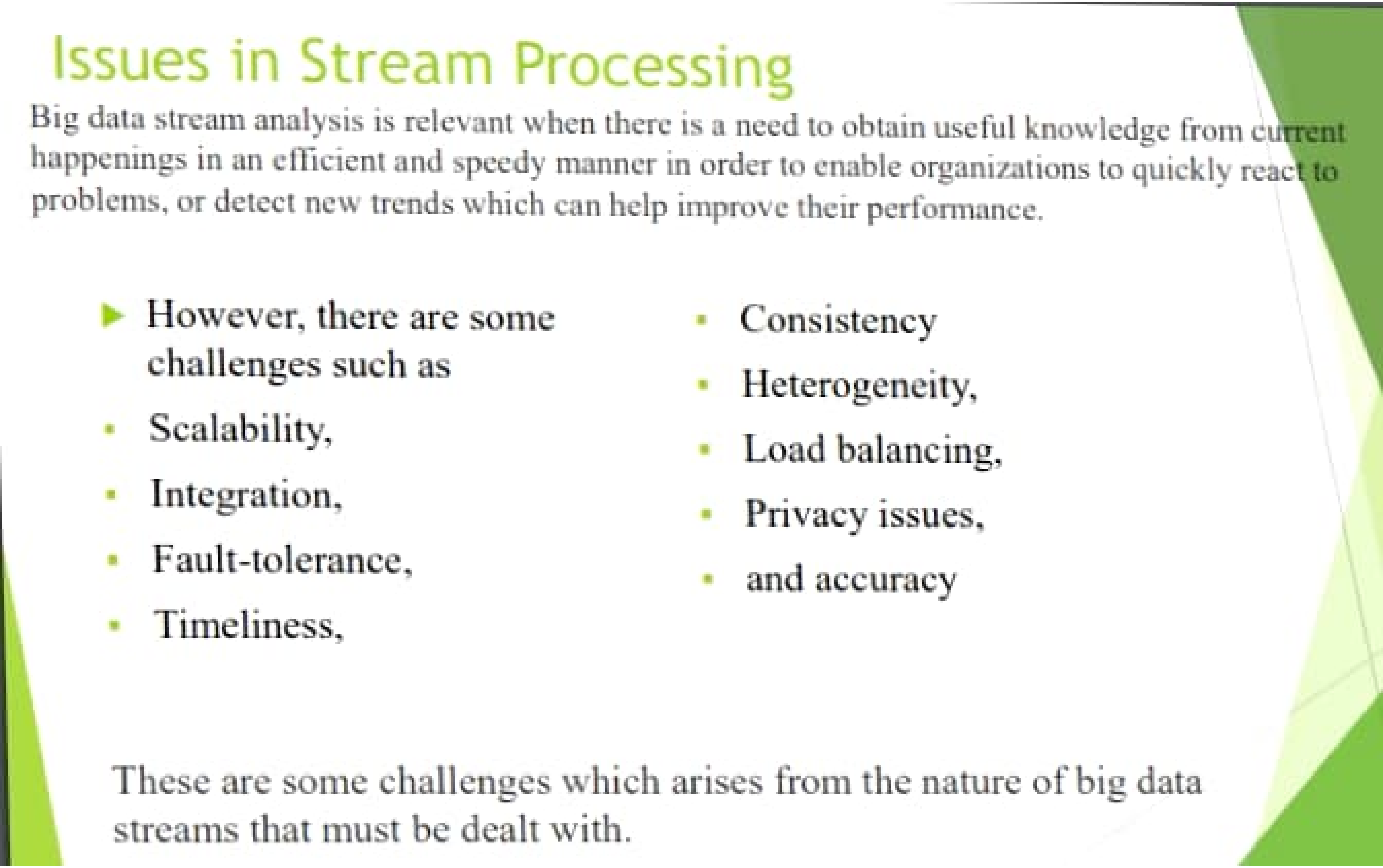
4)

4) Math fuction :

R provides th Various mathematical fuction ito perform th mathematical Calculation. These mathematical function are very helpful to. find absolute Value, Square Value and much more Calculation.

5) String Function :-

R provide Various String function to perform tasks. These String functions allow. us to extract Sub String from String. Search pattern etc. There are the following String function in R.



Issues in Stream Processing

Big data stream analysis is relevant when there is a need to obtain useful knowledge from current happenings in an efficient and speedy manner in order to enable organizations to quickly react to problems, or detect new trends which can help improve their performance.

- However, there are some challenges such as
- Scalability,
- Integration,
- Fault-tolerance,
- Timeliness,
- Consistency
- Heterogeneity,
- Load balancing,
- Privacy issues,
- and accuracy

These are some challenges which arises from the nature of big data streams that must be dealt with.

5.1.2 Examples of Stream Sources

Q. List and explain various data stream sources.

Following are some of the primary sources of stream data :

1. **Sensor data**

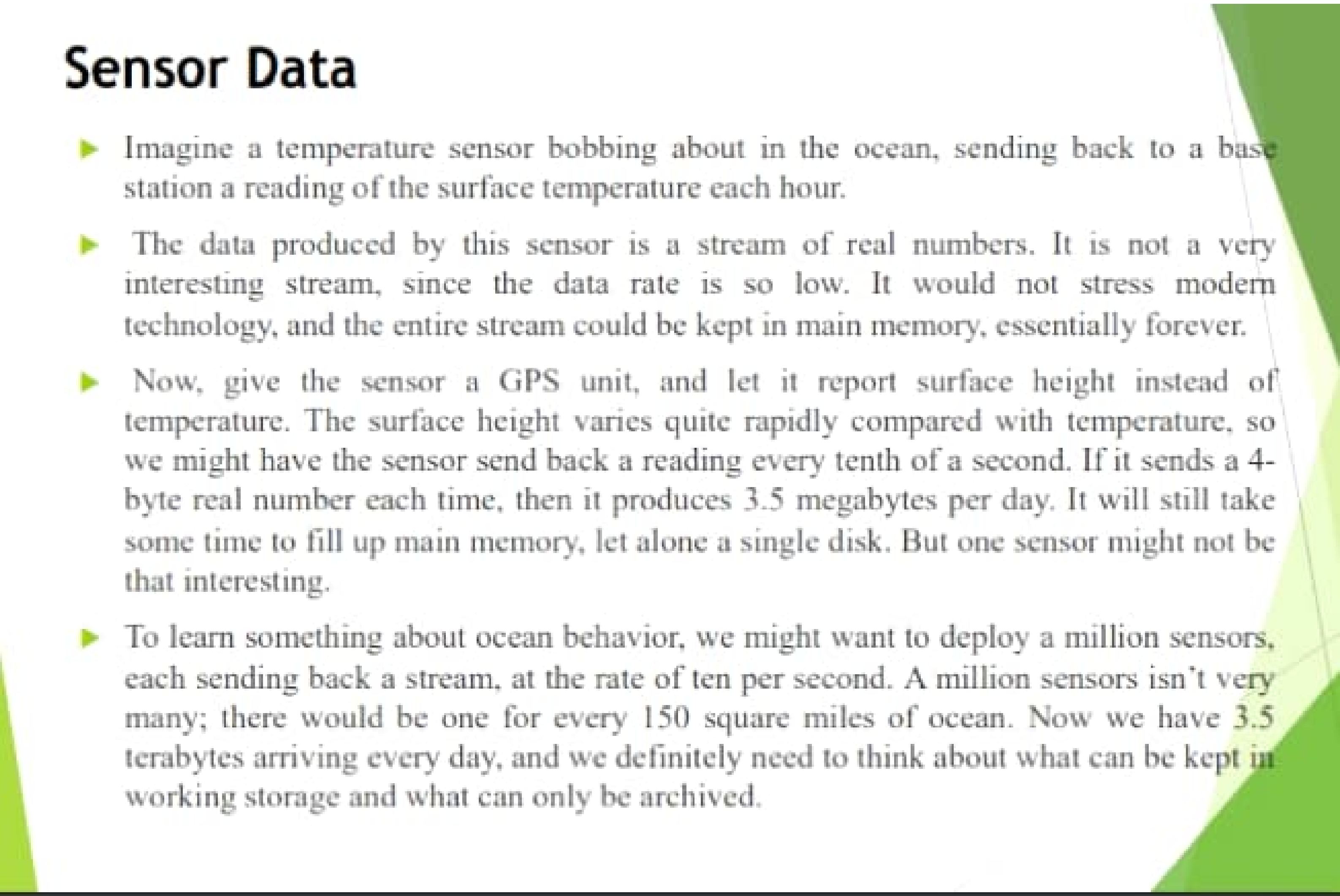
- Sensors are the devices which are responsible for reading and sending the measurements of various kinds of physical parameters such as temperature, wind speed, pressure, moisture content, humidity, pollution level, surface height, amount of light, etc.
- Consider a network of millions of sensors on the ocean surface to provide an early warning system for Tsunami by studying the behaviour of the ocean. This can save a lot of lives. But the amount of data received from all the sensors combined is simply huge.

2. **Data streams related to images**

- High resolution image streams are relayed to the earth stations by the satellites which can amount to many terabytes of image data per day. Many such high-resolution images are released for the public from time to time by NASA as well as ISRO.
- Lower resolution image streams are produced by the CCTV cameras placed in and around important places and shopping centres. Now a days most of the public places and some of the private properties are under the surveillance of CCTV cameras 24×7.

3. **Internet services and web services traffic**

- The networking components such as switches and routers on the Internet receive streams of IP packets and route them to the proper destination. These devices are becoming smarter day by day by helping in avoiding congestion and detecting denial-of-service-attack, etc.
- Websites receive many different types of streams. Twitter receives millions of tweets, Google receives tens of millions of search queries, Facebook receives billions of likes and comments, etc. These streams can be studied to gather useful information such as the spread of diseases or the occurrence of some sudden event such as catastrophe.



Sensor Data

- Imagine a temperature sensor bobbing about in the ocean, sending back to a base station a reading of the surface temperature each hour.
- The data produced by this sensor is a stream of real numbers. It is not a very interesting stream, since the data rate is so low. It would not stress modern technology, and the entire stream could be kept in main memory, essentially forever.
- Now, give the sensor a GPS unit, and let it report surface height instead of temperature. The surface height varies quite rapidly compared with temperature, so we might have the sensor send back a reading every tenth of a second. If it sends a 4-byte real number each time, then it produces 3.5 megabytes per day. It will still take some time to fill up main memory, let alone a single disk. But one sensor might not be that interesting.
- To learn something about ocean behavior, we might want to deploy a million sensors, each sending back a stream, at the rate of ten per second. A million sensors isn't very many; there would be one for every 150 square miles of ocean. Now we have 3.5 terabytes arriving every day, and we definitely need to think about what can be kept in working storage and what can only be archived.

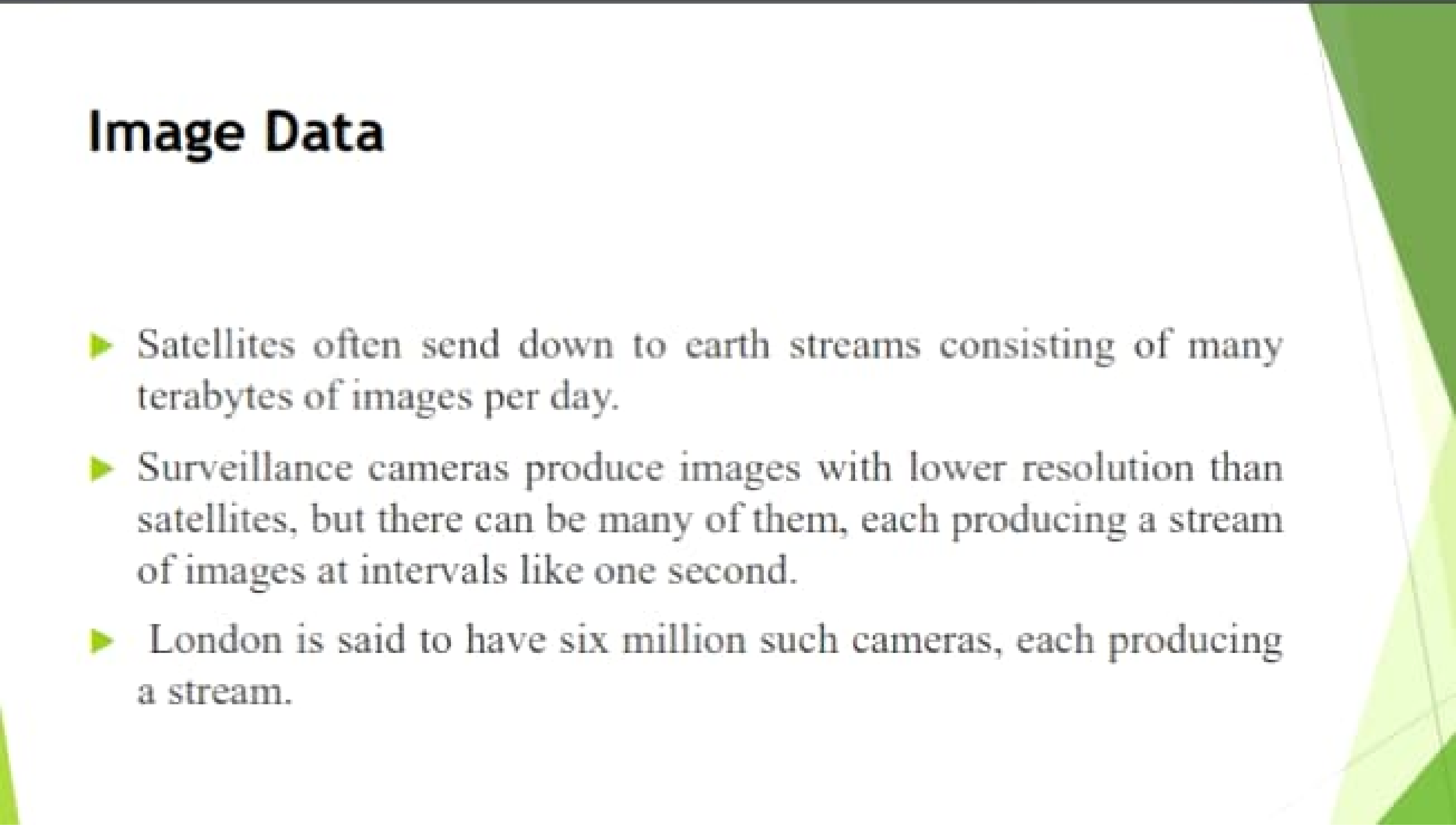
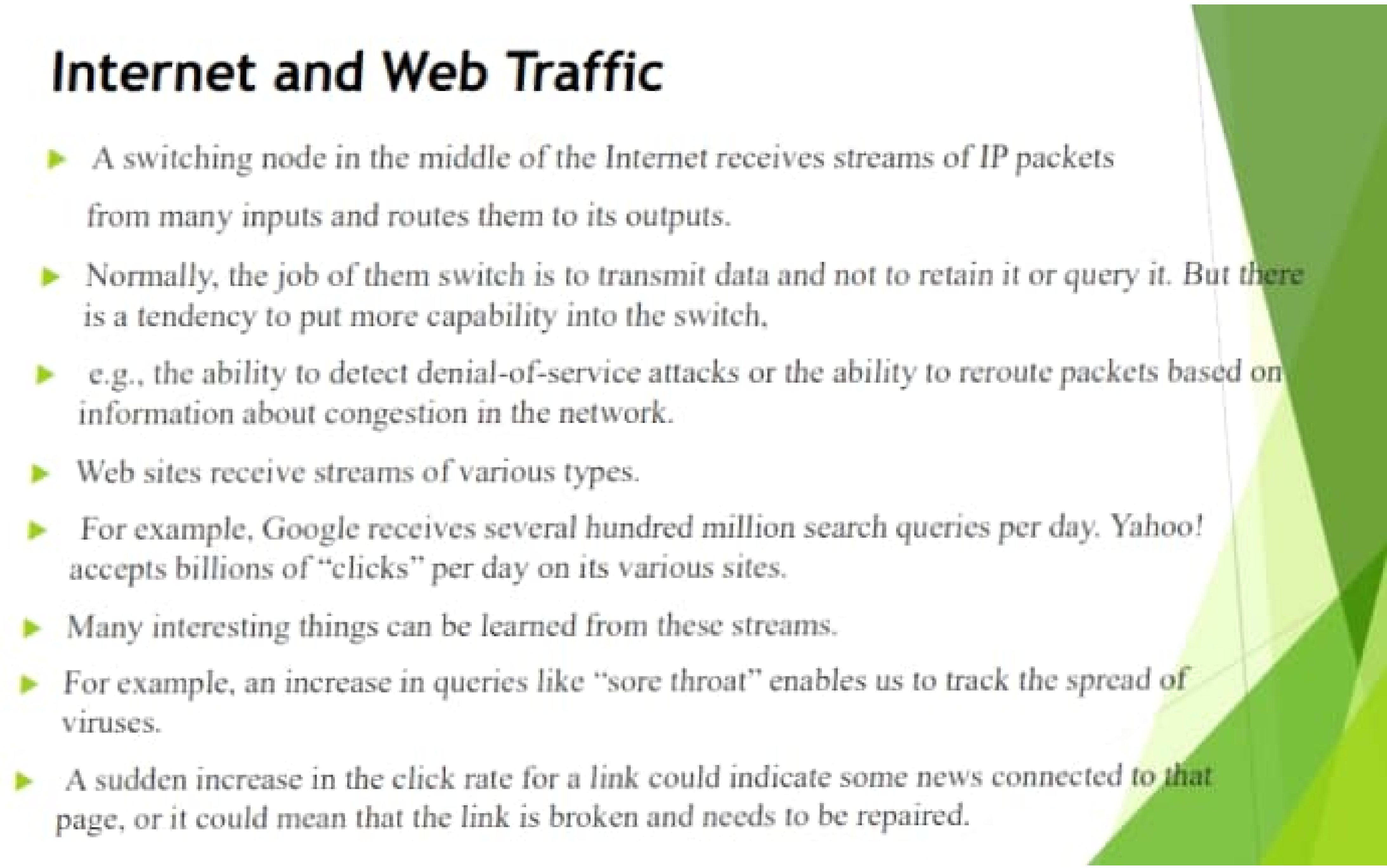


Image Data

- Satellites often send down to earth streams consisting of many terabytes of images per day.
- Surveillance cameras produce images with lower resolution than satellites, but there can be many of them, each producing a stream of images at intervals like one second.
- London is said to have six million such cameras, each producing a stream.



Internet and Web Traffic

- A switching node in the middle of the Internet receives streams of IP packets from many inputs and routes them to its outputs.
- Normally, the job of them switch is to transmit data and not to retain it or query it. But there is a tendency to put more capability into the switch.
- e.g., the ability to detect denial-of-service attacks or the ability to reroute packets based on information about congestion in the network.
- Web sites receive streams of various types.
- For example, Google receives several hundred million search queries per day. Yahoo! accepts billions of "clicks" per day on its various sites.
- Many interesting things can be learned from these streams.
- For example, an increase in queries like "sore throat" enables us to track the spread of viruses.
- A sudden increase in the click rate for a link could indicate some news connected to that page, or it could mean that the link is broken and needs to be repaired.

~~Q15~~) ~~Cosine Similarity →~~

Q12) Explain Flajolet Algorithm with Example ?

→ 1) It approximates the numbers of unique objects in a stream or a database in one pass.

2) Will will remove hash Value, Remainder and $r(a)$

3) Estimation = 2^R

4) Take an Example with Stream

Stream: 4, 2, 5, 9, 1, 6, 3, 7

Hash function: ~~$h(x) = (ax+b) \bmod$~~

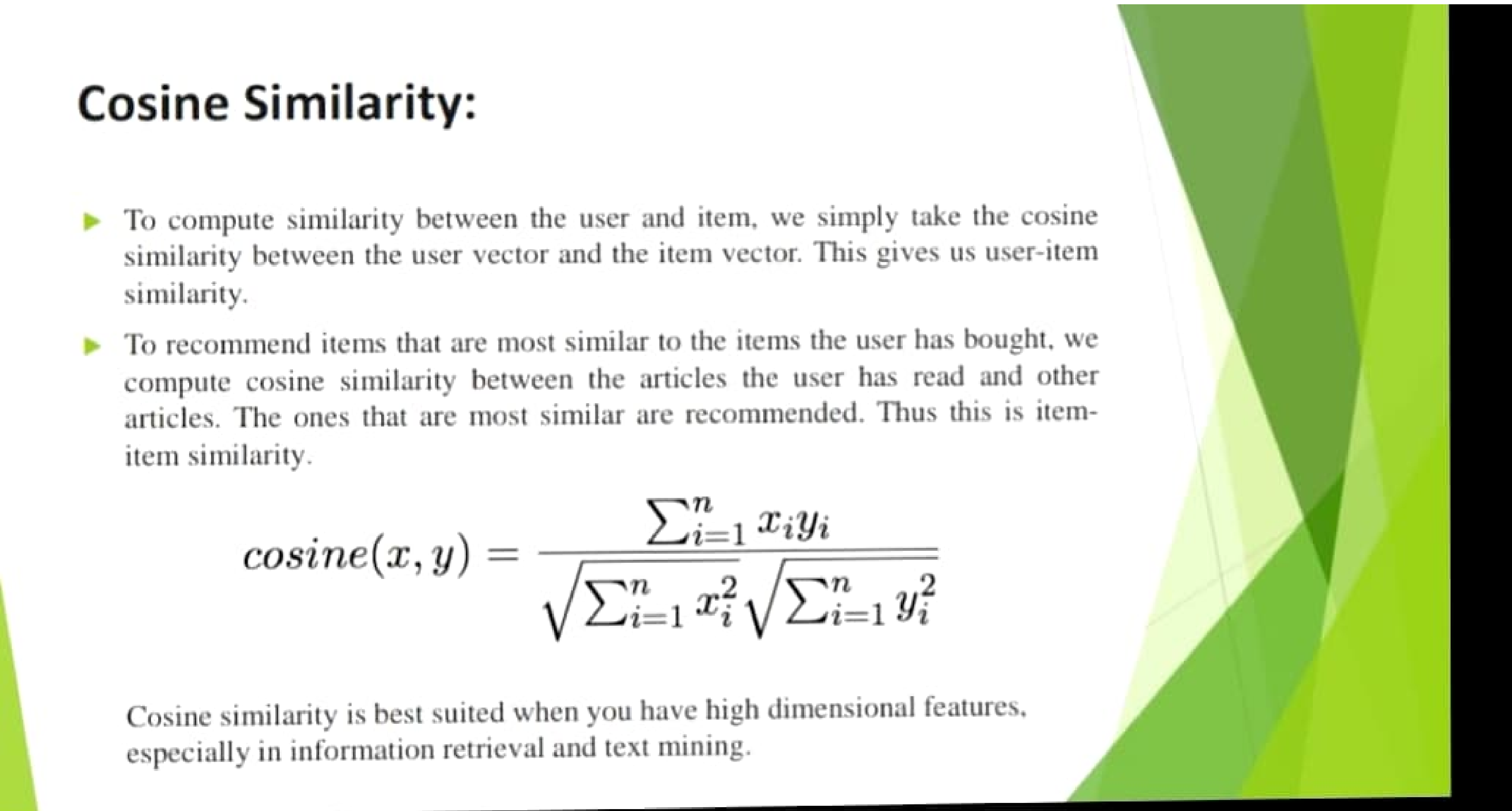
= ~~$h(x) = (6x+7) \bmod 5$~~

$h(x) = 3x+7 \bmod 32$.

$h(4) = 3(4)+7 \bmod 32$ $\quad$ $19 \bmod 32 = 19 = 10011$

x	h(x)	Rem	Binary	r(a)
4	19 mod 32	19	10011	0
2	13 mod 32	13	01101	0
5	22 mod 32	22	10110	1
9	34 mod 32	2	00010	1
1	10 mod 32	10	01010	1
6	25 mod 32	25	11001	0
3	16 mod 32	16	10000	4
7	28 mod 32	28	11100	2

$2^R = 2^4 = \underline{\underline{16}}$

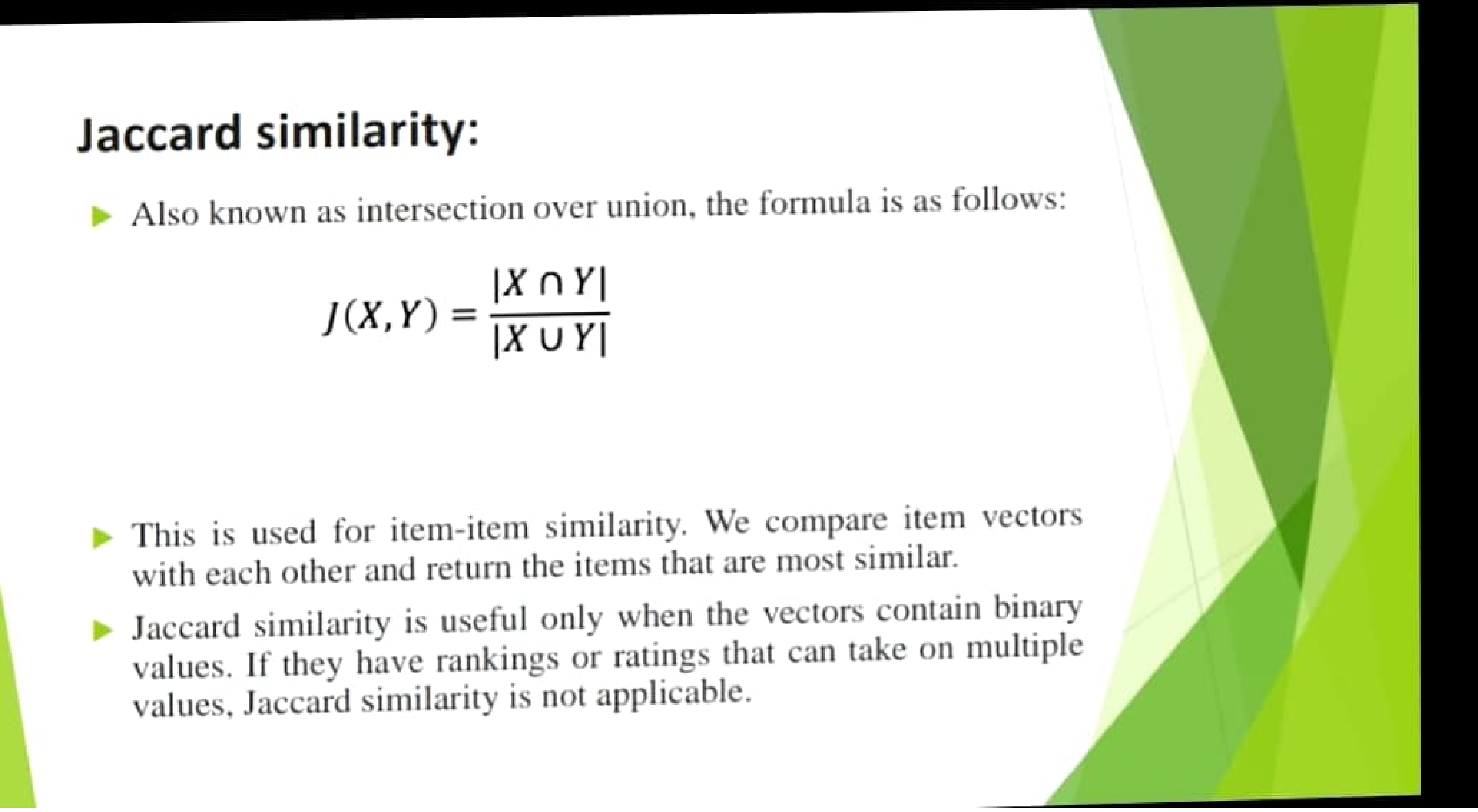


Cosine Similarity:

- To compute similarity between the user and item, we simply take the cosine similarity between the user vector and the item vector. This gives us user-item similarity.
- To recommend items that are most similar to the items the user has bought, we compute cosine similarity between the articles the user has read and other articles. The ones that are most similar are recommended. Thus this is item-item similarity.

$$cosine(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

Cosine similarity is best suited when you have high dimensional features, especially in information retrieval and text mining.



Jaccard similarity:

- Also known as intersection over union, the formula is as follows:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

- This is used for item-item similarity. We compare item vectors with each other and return the items that are most similar.
- Jaccard similarity is useful only when the vectors contain binary values. If they have rankings or ratings that can take on multiple values, Jaccard similarity is not applicable.