

## Specialised storage paradigms

### Blob Storage

Blob  $\Rightarrow$  Binary Large Object (This is an arbitrary piece of unstructured data)

Blob storages are used to store and retrieve massive amounts of unstructured data such as (mb even sometimes GB of data) images, database snapshots, video files, text files and other static assets that a website might have.

Eg: GCS  $\Rightarrow$  Google Cloud Storage

S3  $\Rightarrow$  AWS, Azure Blob storage

Even though Blob storage is very similar to a key-value storage (because a blob is a blob storage is accessed using a key) they are optimized for different use cases (In a blob storage we can effectively store and retrieve mb or even GB of data but might be difficult in a key-value storage)

## Time Series Database

This is used for **storing** and analysing **time-indexed data** (Eg: For logging and monitoring system performances, Stock price app (price changes with respect to time), **IoT devices**)

Eg: ① Influx DB      ② Amazon TimeSeries  
② Prometheus  
③ Graphite

## Graph DB

A graph database is built on top of the graph data structure.

Graph databases are more preferred to relational databases when the **data points** have **multiple level of relationships**. Eg: Social Network

Thus the main advantage of graph database is to perform complex queries on deeply connected data.

Ex: Amazon Neptune

Neo4J → Cypher : This is a graph query language

## Spatial Database

A type of database optimized for storing and querying spatial data like locations on a map. Spatial databases rely on spatial indexes like quadtrees to quickly perform spatial queries like finding all the locations in the vicinity of a region.

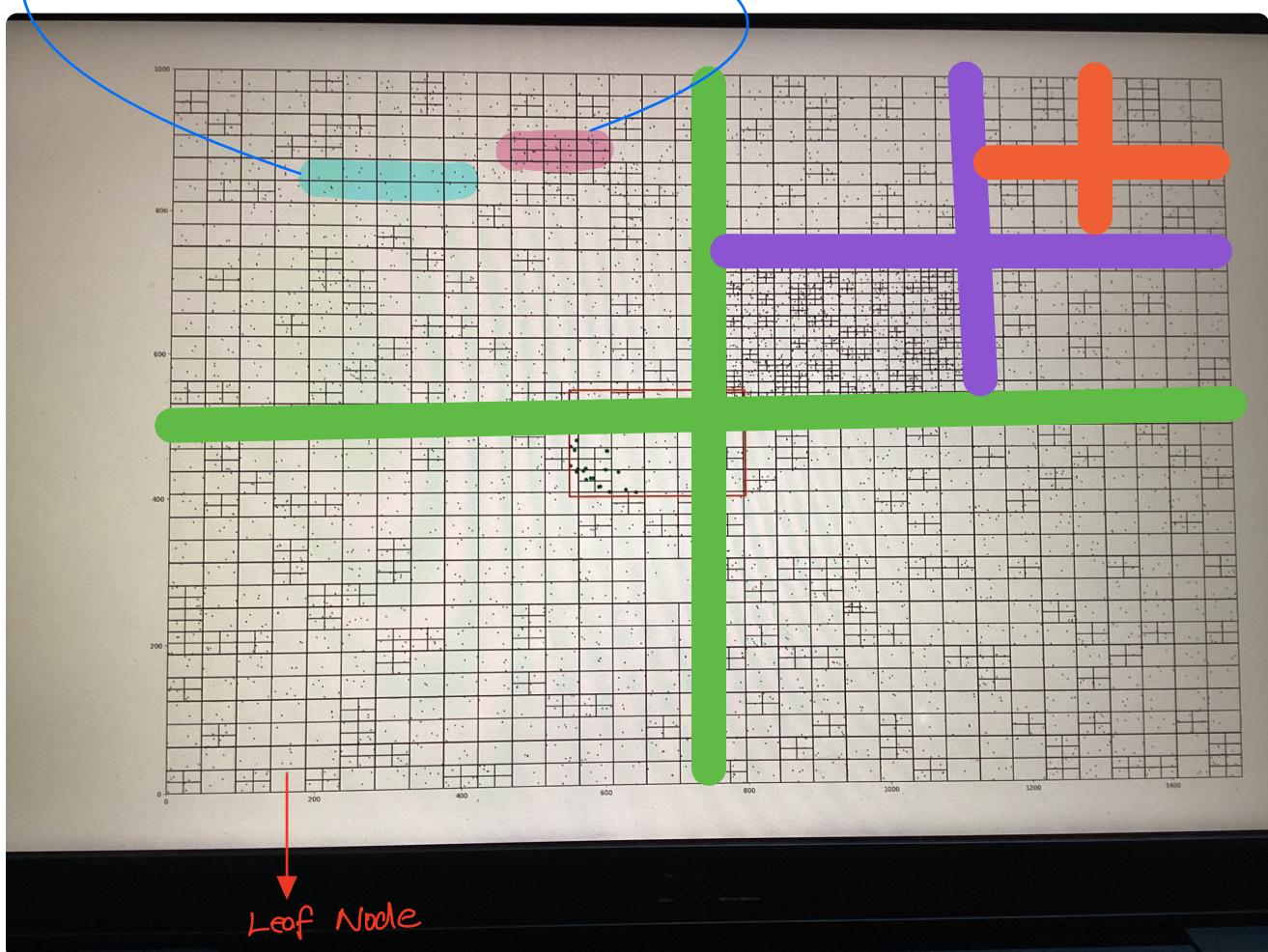
### Quadtrees:

Just like a binary tree has two nodes, a quadtree has exactly four nodes or zero nodes (and is therefore a leaf node).

As long as the nodes aren't at capacity they remain leaf nodes, once they reach capacity, they are given four children nodes and their data entries are split across the four children nodes.

Eg: If a node can hold upto 5 entries, if sixth entry is inserted it will be split into four children nodes.

These boxes highlighted in blue are not subdivided because they have lesser number of data entries compared to the boxes in red.



→ Here the outermost rectangle is the root node

→ Finding a location in a quadtree is an extremely fast operation  $\Rightarrow \log_4(n)$

where ' $\omega$ ' is ' $V$ ' the total number of locations  
 $\log_4 \Rightarrow$  since quarantines have four children nodes