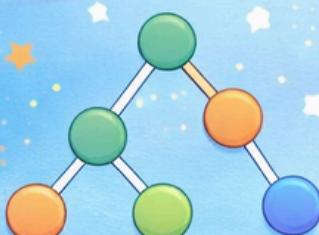




# Sriju's Super Fun Guide to



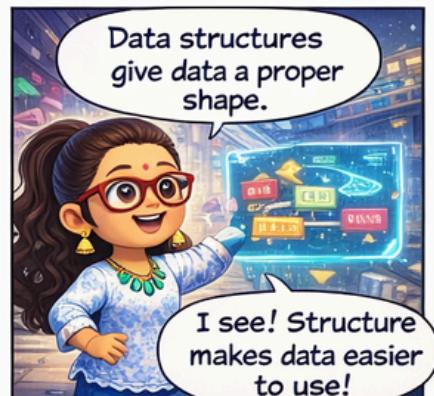
# DATA STRUCTURES



## EPISODE 1

# Sriju Enters the DATA STRUCTURES FACTORY

Whoa...  
where am I?



## EPISODE 2

# The ARRAY ASSEMBLY LINE

Arrays... I've heard this word before.



Data moves in a straight, ordered line.



So every item has a number?

Yes. Arrays use index numbers.



If I know the index, I can get data quickly!



What if I add one here?



Arrays work well for fixed lists.

So arrays are good!



Next department: Stack Storage Tower

### EPISODE 3

# The STACK STORAGE TOWER

This place looks... tall.

Top only!

Data is stored vertically here.

This is a stack.

We add data on the top.

And remove from the top.

I'm pushing data.

push

Last in, first out!

I'm pushing data.

I'm popping data.

pop

## EPISODE 4

# The Queue Dispatch Zone

I made it to the Queue Dispatch Zone!

Data lines up in the order it arrives.

Queues follow FIFO.

First in... first out?

FIFO

FIFO

First In → First Out

Enqueue puts data at the back of the line.

First in, first out.  
Order matters.

enqueue

dequeue

• Next stop:  
Hash Vaults

## EPISODE 5

# The HASH VAULTS

Wow... this place looks important!

HASH VAULTS :



Each vault stores data safely.

Welcome to the Hash Vaults.



How do we know which vault to open?



## EPISODE 6

# The HASH VAULTS

## FAST LOOKUP

There are so many vaults! Do we search them one by one?



We convert keys into numbers.



Keys are converted into numbers.

hash("apple") → 4

Index generated



That was super fast!

THE SYSTEM JUMPS DIRECTLY TO THE VAULT.



That's why hash tables are efficient.

## EPISODE 7

# The HASH VAULTS

### -- COLLISIONS & SOLUTIONS --

What if two keys choose the same vault?

COLLISION ZONE!



Both keys map to the same index.

hash("berry") → 5

-hash("ruby")→ 4

Index generated

4

Collision detected!

berry      ruby



This is a collision!!



Collisions can happen. We handle them.

## EPISODE 8

# The HASH VAULTS

- REPLACING HASH TABLES -

What if two keys choose the same vault?

COLLISION ZONE



Links are updated.

hash("berry") → 5

hash("ruby") → 5

Index generated

4

Adding is easy.



No shifting needed!

Collisions can happen. We handle them.



Next: Episode 7 - When to Use Hash Tables

## EPISODE 9

# When USE HASH TABLES

### - WHEN TO USE HASH TABLES -

Hash tables are really fast.

Keys lead directly to values.



### When NOT to Use Them

Hash tables do not keep order.

Hash tables do not keep order.



Other structures handle order better.



# LINKED LIST

- WHEN AND WHY TO USE -

This is a linked list.



A linked list is a collection of connected nodes.

Each node has two parts.



There are no index numbers.

To reach data, we move step by step.



No shifting is required.

Adding data is easy.

Searching takes more time.

- ✓ Flexible size
- ✓ Easy insert and delete
- ✗ Slower access than arrays

**A Sriju Comic**



**Curiosity Lives Here!**

**Stories that smile back at you**