**HASHING**

Store data in key-value pair

It used to reduce time complexity in searching and accessing data

**OBJECTIVE : need to optimize searching time**

Example :- 4 2 6 1 3

How can we store data?

Data Structure searching time insertion time

In an array : o(n) o(1)

Sorted array : o(log(n)) o(n)

(performing Binary Search i.e logn )

Linked List : o(n) o(1)

Balanced BST : o(log(n)) o(log(n))

**But we want searching and insertion in o(1)**

**Hashing:** Now we still store data in an array but in non sequence manner

Like lets store data in arr[i]=i

Here searching and sorting is in o(1);

DRAWBACK: lets say need to store data :- 3 2 1 4000

Here we need array of size 4001 i.e lots of memory is wasted

Lets say a better approach :- lets pass an integer in HA(Hashing function) and give index to store data

Lets say= HA=i%10=index;

To store 3 => 3

To store 45 =>5

To store 1 =>1

To store 4000 =>0

Now can overcome issue of wasting huge memory but there is another drawback->

How to store 1,11,21,31 etc.

This is called collision

Collision: Single key mapping to multiple values