INTRODUCTION TO DATA STRUCTURE

WHY DATA STRUCTURE?

- Data Structure is the most fundamental and building block concept in computer science.
- Good knowledge of data structures is must to design and develop an efficient software system.
- Data structures can be defined as the process of collecting and organizing data in the best way.

There are mainly two types of data structures

- Primary data structure: Integer, float, Boolean, char, long, etc. are known as primitive data types. The concept of handling primitive data types in an efficient way is known as Primitive Data structure.
- Abstract data structure: in programming and businesses, we also have to use data types such as stack, linked list, graph, queue, tree, etc. These kinds of data types are known as complex data types. Abstract Data Structure is the technique or concept of handling the connected, complex, and large amount of data in an efficient way.

Revisiting dictionaries and sets

- Before revisiting dictionaries and lists we need to talk a bit about hash maps.
- Hash maps are indexed data structures. A hash map makes use of a hash function to compute an index with a key into an array of buckets or slots.
- Let's discuss what is the hash function.

HASH FUNCTION

- A HASH FUNCTION MAPS A BIG NUMBER OR STRING TO A SMALL INTEGER THAT CAN BE USED AS THE INDEX IN THE HASH TABLE (KEY).
- A PRACTICAL HASH FUNCTION SHOULD UNIFORMLY DISTRIBUTE DATA, AND IS NOT COMPUTATIONALY EXPENSIVE.

EXAMPLE OF A HASH FUNCTIONS

- The mod function
- In this method for creating hash functions, we map a key into one of the slots of table by taking the remainder of key divided by table_size.
- since it requires only a single division operation, hashing by division is quite fast.
- It has been found that the best results with the division method are achieved when the table size is prime.

COLLISIONS

- Suppose **R** = 256 and **TABLE_SIZE** = 17, in which R % TABLE_SIZE i.e. 256 % 17 = 1.
- So for **KEY = 37599**, its hash is 37599 % 17 = 12
- So for **KEY = 37599**, its hash is also 12
- Hence it can be seen that by this hash function, many keys can have the same hash. This is called collision
- Can be solved in many ways , open addressing uses for example linear probing.

Summary

- Hash function is the core of implementing a hash map.
- It takes in the key and translates it to the index of a bucket in the bucket list.
- Ideal hashing should produce a different index for each key.However, collisions can occur.
- When hashing gives an existing index, we can simply use a bucket for multiple values by appending a list or by rehashing.

Continued

- In python dictionaries are an example of a hashmap.
- Set as well is also considered hash map.
- Both of them are o(1) (concept of big o notation will be discussed later on).
- Dictionaries and sets are fast because of this fact.

EXAMPLE CODE

- The hash map design will include the following functions:
- **set_val(key, value):** Inserts a key-value pair into the hash map. If the value already exists in the hash map, update the value.
- **get_val(key):** Returns the value to which the specified key is mapped, or "No record found" if this map contains no mapping for the key.
- **delete_val(key):** Removes the mapping for the specific key if the hash map contains the mapping for the key.

Difference between lists and arrays

- We have a concept called linked lists and doubly linked lists.
- Each element in a linked list points to the other , no need for them to be saved consecutively.
- Arrays are saved consecutively so it is faster.
- Lists in python take the best of both worlds.