Hashed Data Structures

A **Hash Table** is a data structure that can be thought of as an array. Its number one advantage over alternative data structures is that it allows for both very fast Insertion and Searching operations. Key values are assigned to elements in a Hash Table using a Hash Function. **Hash Functions** calculate the best index that a value should be stored in. Because these data structures are based off of arrays, they are limited in size, and therefore the index calculated by the Hash Function must be small enough to fit inside of the array/structure size. Also, it cannot overwrite values already in the structure. Hash functions store values in the structure in a way that the array does not need to be searched in order to find the correct index. Instead, values can be entered in any order, and then can be found by using a calculation. This is what makes Hashed Data structures so fast, and the reason for that is because memory accesses are slow, whereas calculations are very fast.

For example:

Say you have a piece of information stored in a Hash Table that is identified with a unique, nonduplicate ID value. This ID is calculated in the Hash Function, and the result of this calculation provides the exact index for the information with said ID. It then goes directly to that index in the array / Hash Table, and then returns this information to the user.

Hash Table 🡪 is a Data Structure (Think of it as an array)

* Fast Insertion & Searching
* Limited in size (since they are based off an array)
* Can be resized BUT should be avoided
* Hard to order

\*Key Values are assigned to elements in a Hash Table using a Hash Function\*

Hash Function 🡪 calculates the best index an item should go in

* Achieves the task of storing values in an array with a limited size, in a way that the array doesn’t need to be searched through to find it
* Allows you to enter values in any order
* It then finds the values using a calculation instead of searching 🡪 hence VERY FAST
* Index must be small enough for the array size
* Don’t overwrite other data in the Hash Table

Hash Functions == Speed

* Nonduplicate ID for info stored in a Hash Table
* Calculation(ID) 🡪 provides the exact index where that info is located in the Hash Table (array)
* Goes directly to that location in the structure and returns/sends that info to the user

hashFunction1

hashFunction2

* Plan to store values between 0 – 999 BUT never plan to have more than 15 values stored in each array 🡪 wasteful to use the first function
* Use the Mod function 🡪 take the modulus (remainder)

\*Collision: trying to insert a value into an index that already contains a value