Data Structure

Arrays -> Stack -> Queue -> LinkedList -> Trees -> Graphs

Interview Prep

Arrays -> Hashmaps -> Stack -> Linked List -> Trees -> Queue -> Graphs

-> Likelihood of the Questions you can face:

Essentials : Arrays, Strings, Hashmaps

-> HASHMAP   Key & Element values

Unique {
```
[  $H>1 |
 1<2->x-2
 1<3> :
      .
      .
      .
```
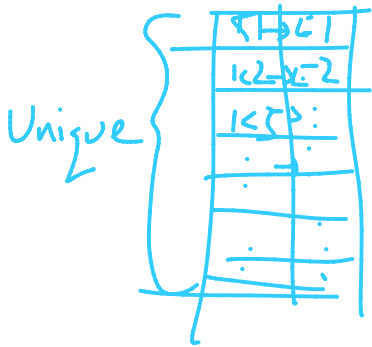
{ key : elements }

Python : Dictionary

-> Problem - 01 :   Contains Duplicate

[1, 2, 3, 4]   = true

1, 2, 3, 1

{ 1:3, 2:1, 3:1 }

—or loop through array :
i_ value is in the hashmap:
      hm [value] += 1
else:
      hm [value] = 1

```
keys = hashmap.keys()
for keys:
    if hashmap[key] >1
        return true
return false
```

→ Problem 02: Valid Anagram

string 1 = anagram ⟹ chars : counts

string 2 = nagaram ⟹ counts for chars

If any keys are essentially zero

⇒ Declare hashmap

- for loop on string 01:
```
    if s[i] is in hashmap:
        hashmap[s[i]] += 1
    else
        add the element as 1 to the key
```

for loop on string 02:
    decrement the count

keys = [_ - - - -]
           ↑
      Element != 0
        return false

return false

return True

n   Two Sum

[2, 7, 11, 13]   target = 9

for...for $O(n^2)$ $O(n)$ — ~~free~~ Efficient

-> Declare hashmap

-> for key, value in enumerate(inputArr):
    diff = target - value
    if diff is in the hashmap:
        return [hashmap[diff], key]
    hashmap [value] = index

[1

1

9

return