Array

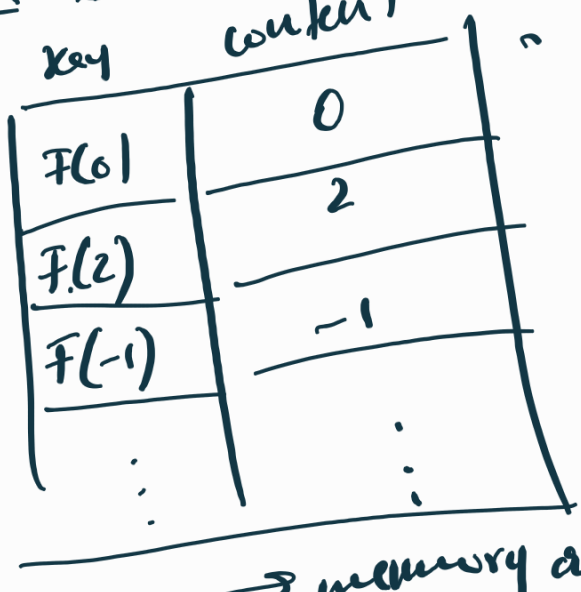
linear search O(n)

index → memory addr

0 → 0xae108fd7

1 → 0xae108ffe

Hash table



content: X

key : F(x)

search 2:

1. Get F(2) (O(1))
2. Go to F(2) O(1)

Search O(1)

key → memory address

F(0) → 0xeef20ed

F(1) → ...

F is a hash function

$F: D \longrightarrow R$

domain — range:

F_x: hash table for strings of size 4

$x \longrightarrow F(x)$

Juan	→ Hex(Ju)	32 bits
Joha	Hex(Jo)	
Paul	Hex(Pa) ←	h(Ju): Juan
Cesa	Hex(Ce)	h(Jo): Joha
Edua	Hex(Ed)	⋮
Andr	Hex(An)	h(Pa): Paul, Pabl
Pabl	Hex(Pa) ←	

$x_1 \neq x_2$ s.t. $F(x_1) = F(x_2)$

We have collision

Hash map
dictionary

{'Ju': 'Juan',
'Jo': 'John',
'Al': 'Alex',}.

key : value
provided by the developer

Python H(key).

Hash set
sets

Z = ['J', 'A', 'L', 'J']

print (Z)

>> [J, A, L, J].

M = {'J', 'A', 'L', 'J'}

print (M)

>> {'J', 'A', 'L'}

unique : array → set.

value provided by the developer

Python H(val)