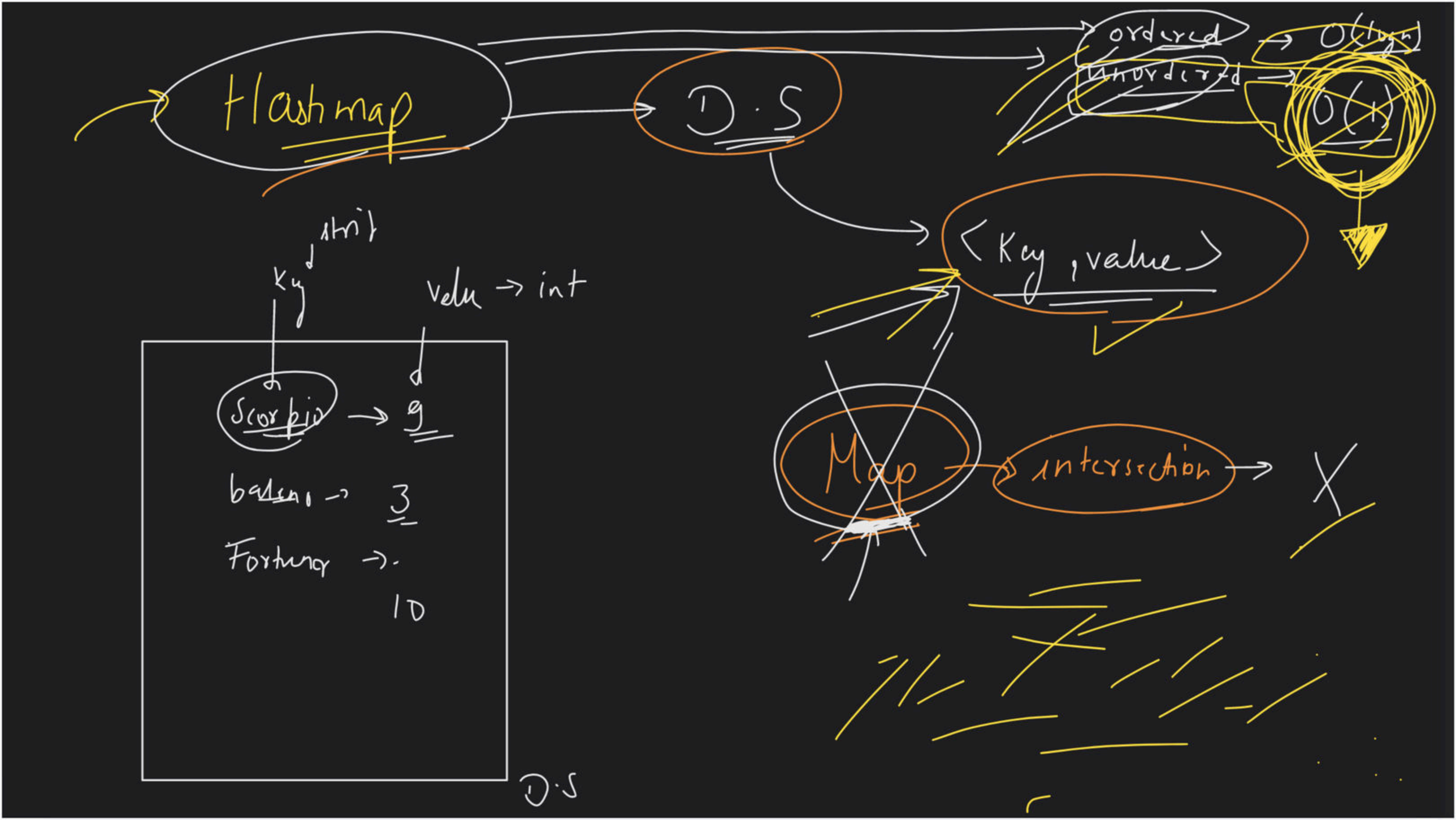
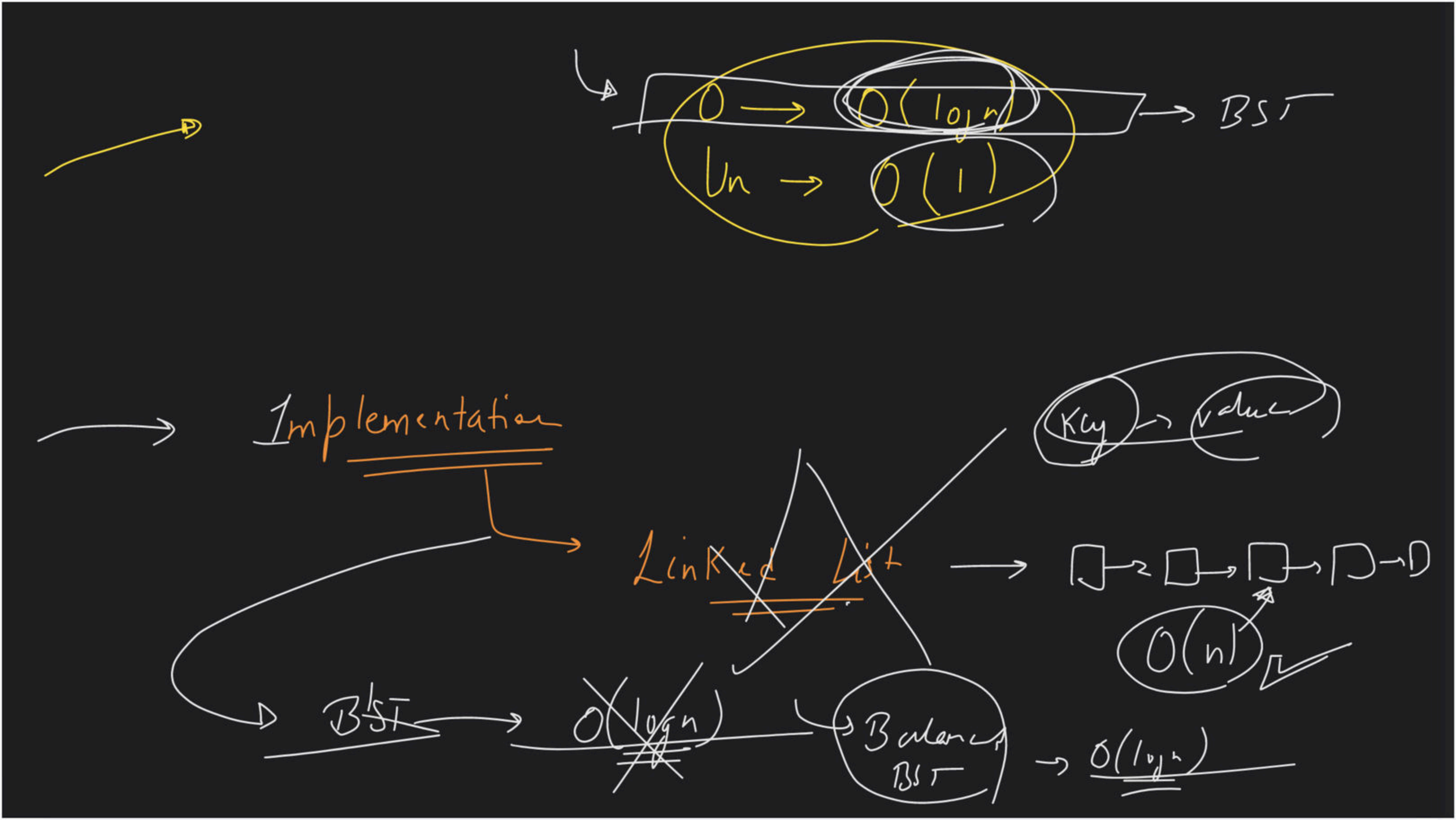
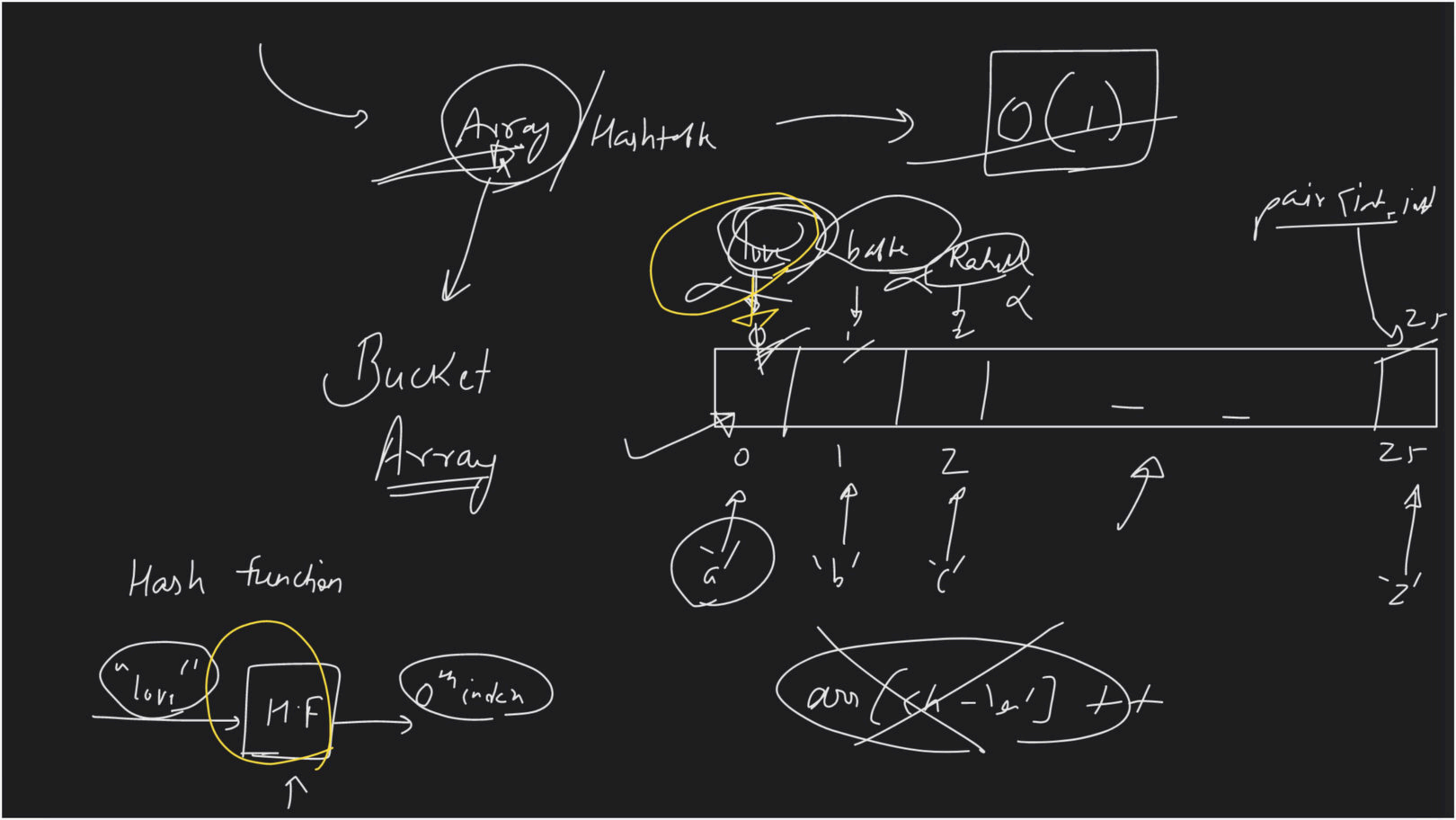
Hashmaps & Tries - Class 1

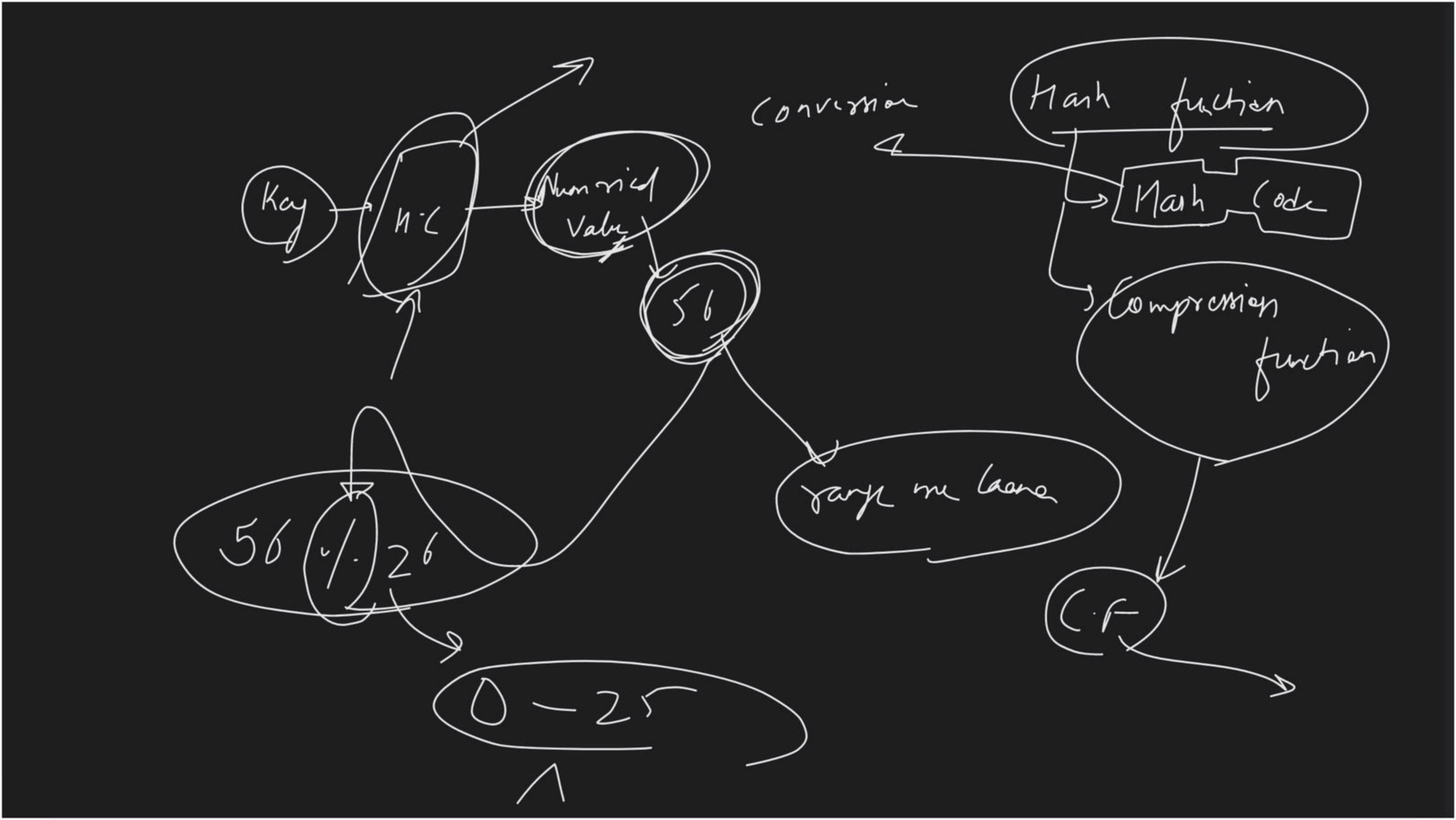
Special class

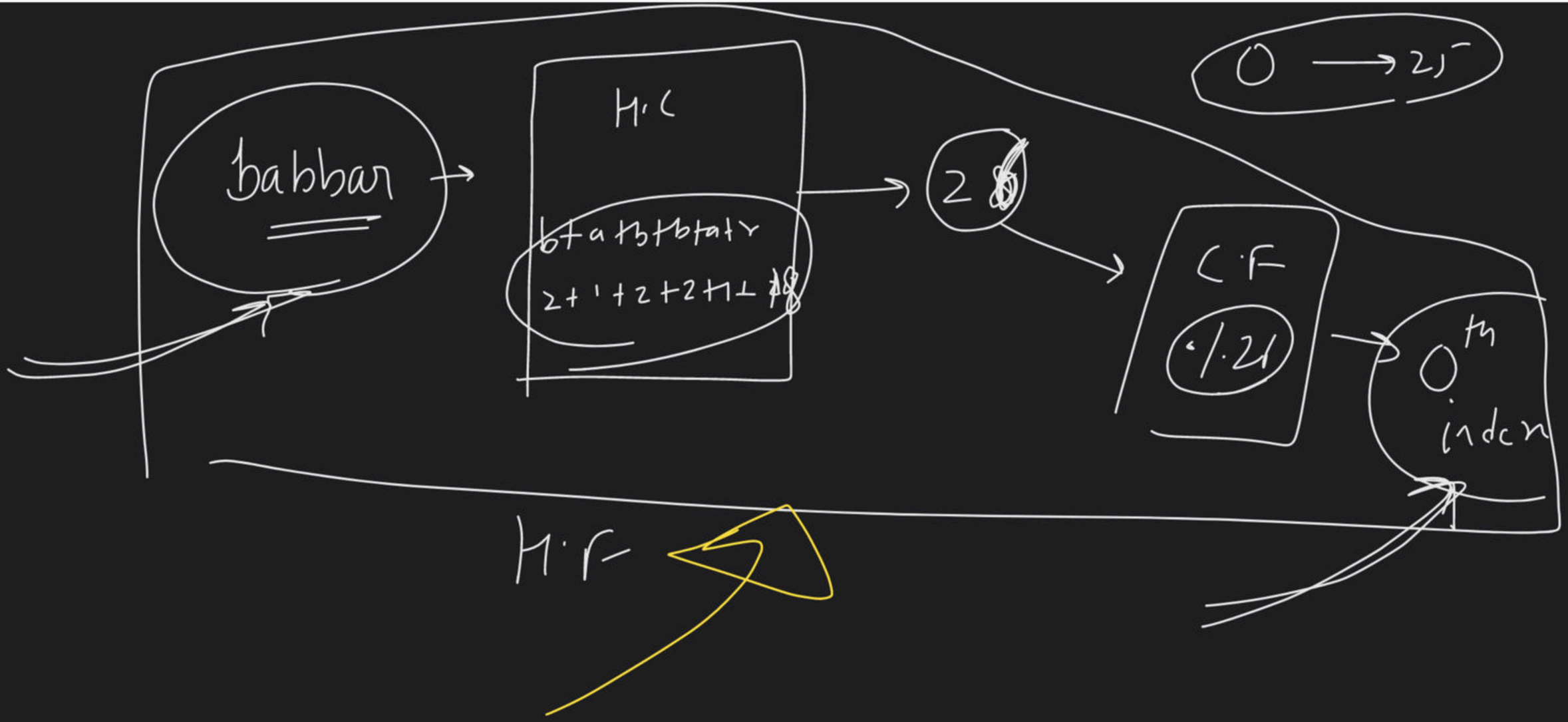
Love Babbar • May 28, 2023

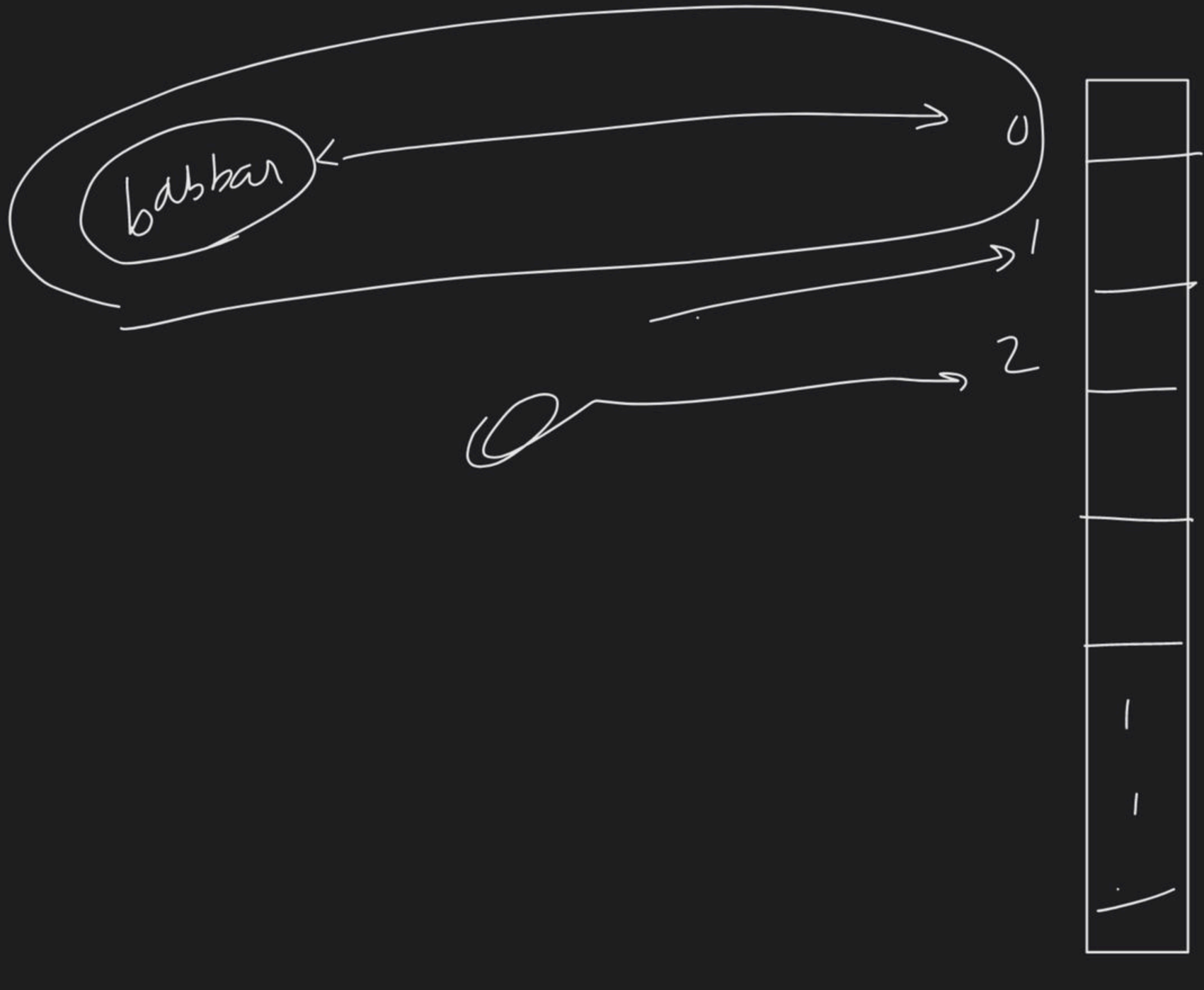

Hashmap
D.S
ordered
unordered
O(log n)
O(1)
<Key, value>
Key
str
value -> int
Scorpio
9
batman -> 3
Fortuna -> 10
Map
intersection
X
D.S

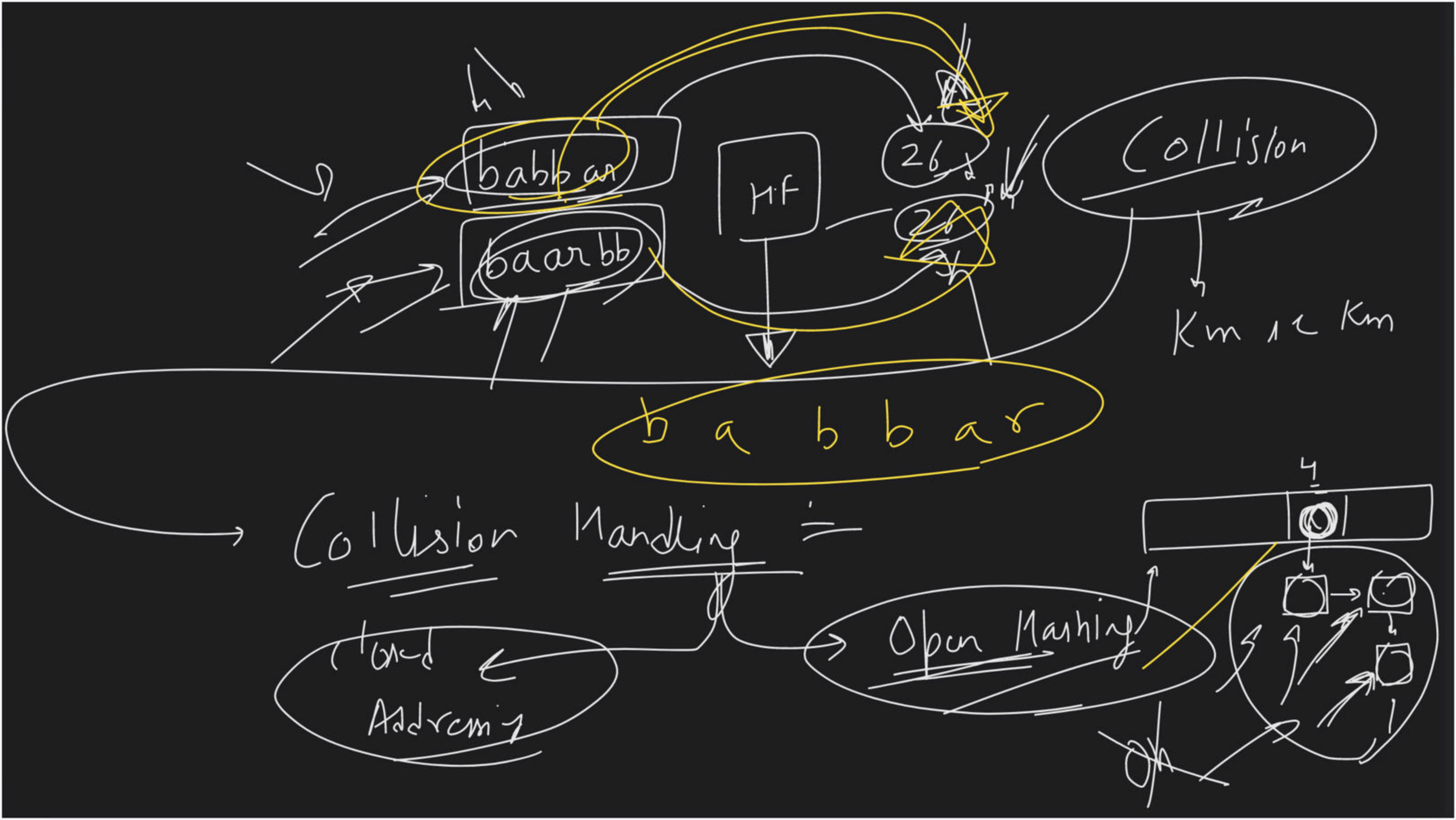

O → O(log n)
Un → O(1)
BST
Implementation
LinKed List
Key → value
O(n)
BST
O(log n)
Balanced BST
O(log n)

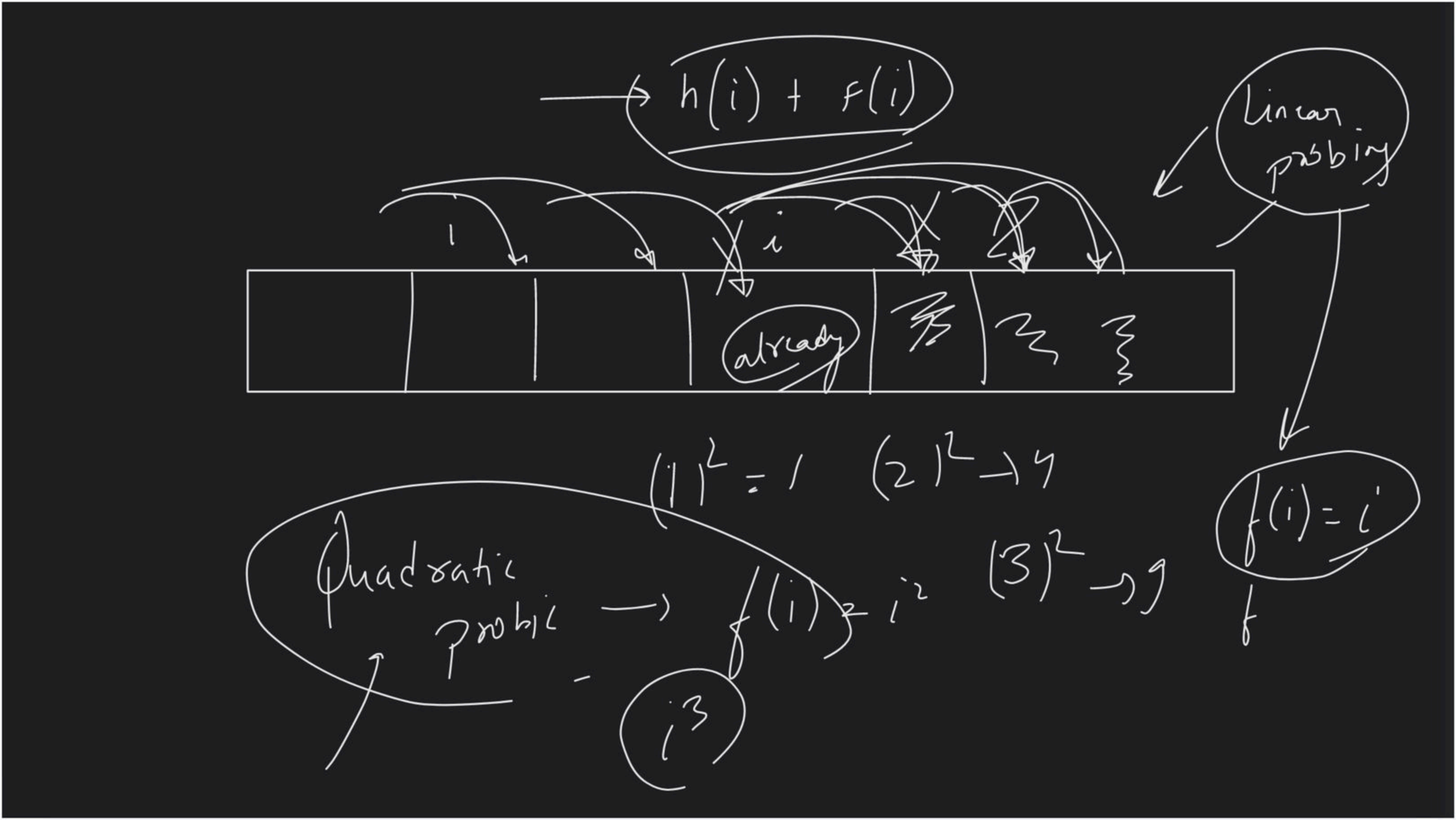

Array / Hashtable
O(1)
Bucket Array
love
babbar
Rahul
pair<int, int>
0
1
2
25
'a'
'b'
'c'
'z'
Hash function
"love"
M.F
0th index
arr['a' - 'a']++

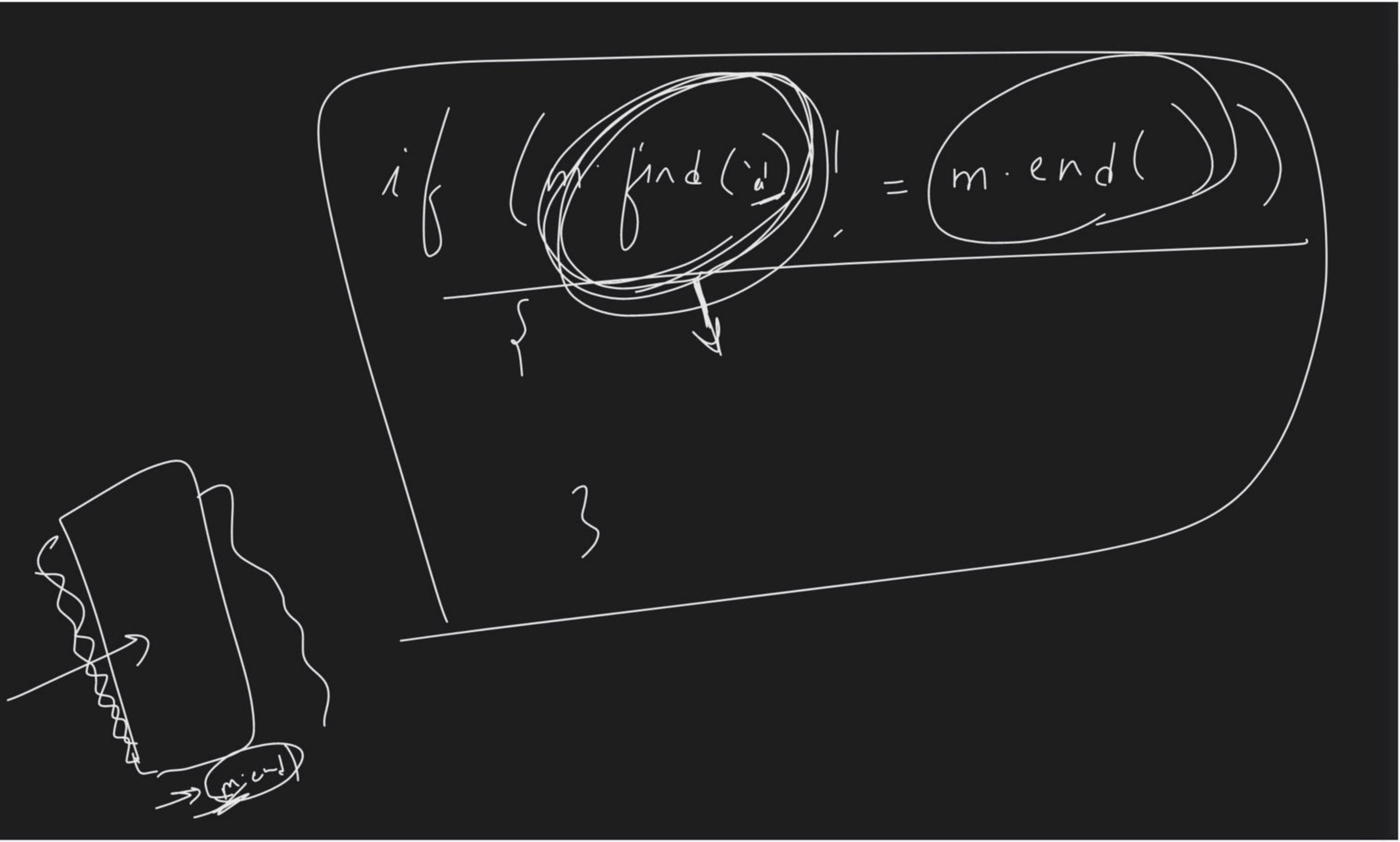

Key
M·C
Numerical Value
56
56 % 26
0 - 25
conversion
Math function
Math Code
Compression function
range me laana
C·F

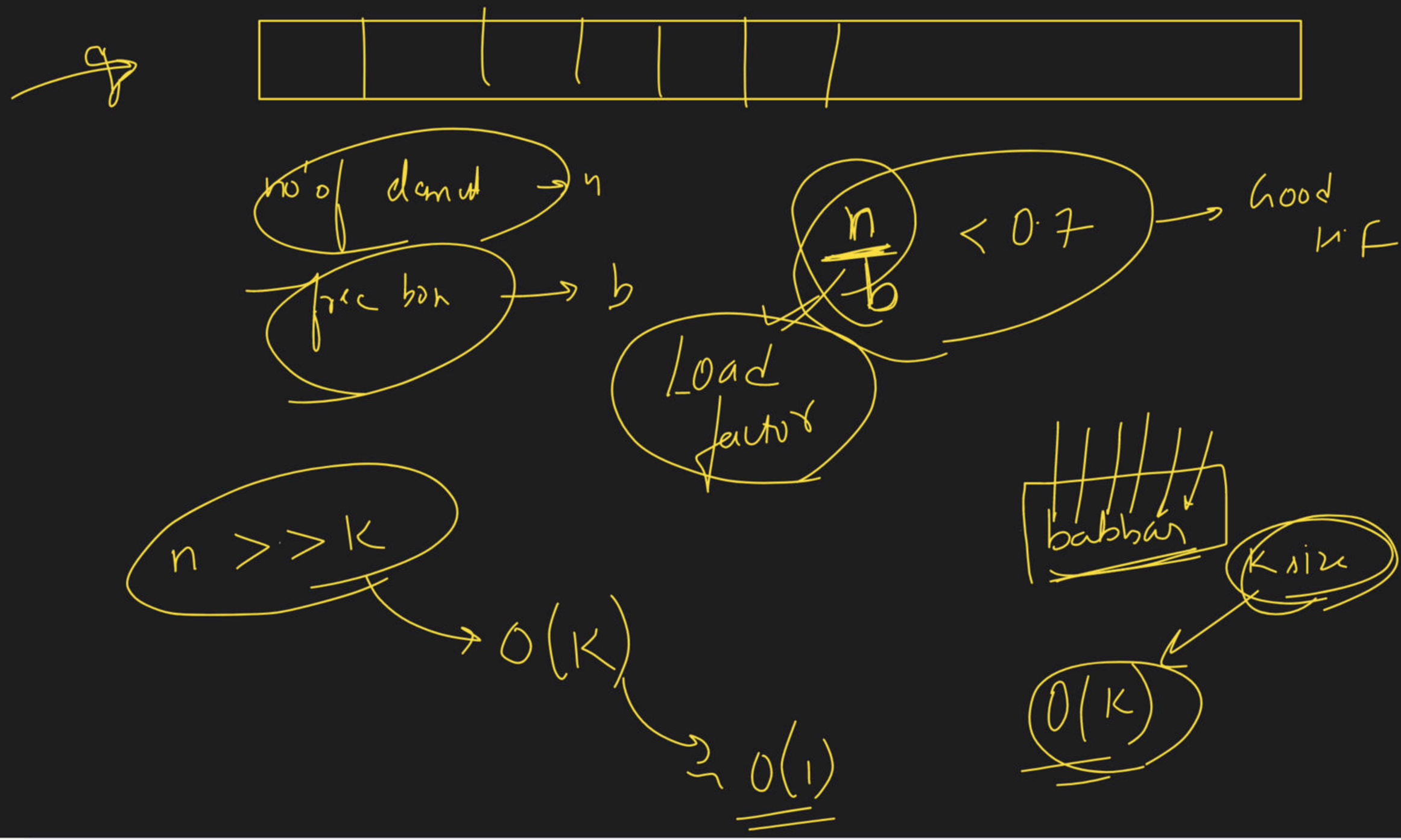

babbar
H.C
2
C.F
1.21
0th
index
H.F

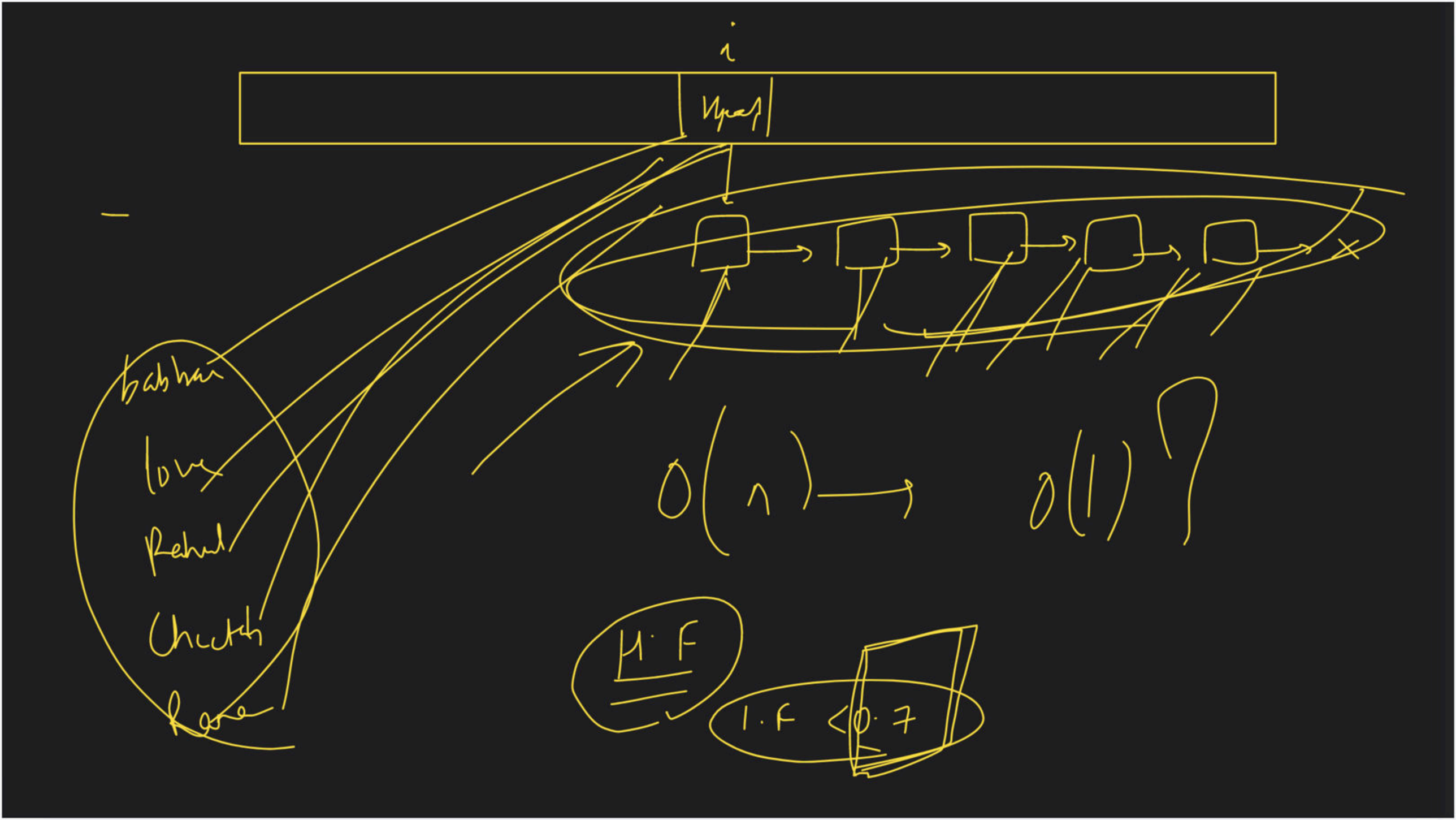

babbar
0
1
2


babbar
baarbb
HF
26
20
Collision
Km 12 Km
babbar
Collision Handling
Open Hashing

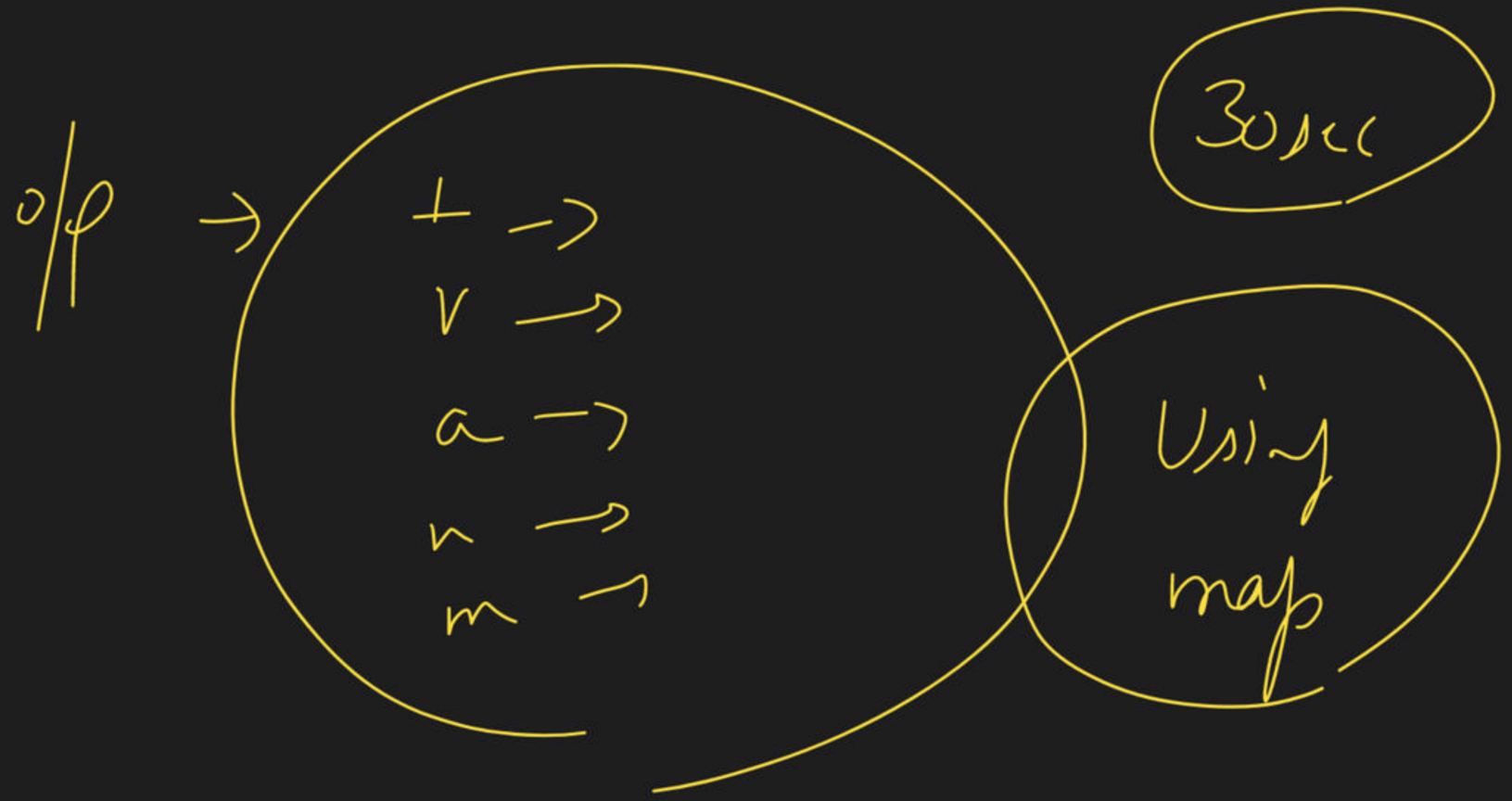

h(i) + f(i)
Linear probing
1
i
already
(1)² = 1
(2)² → 4
(3)² → 9
Quadratic probing
f(i) = i²
i³
f(i) = i
f

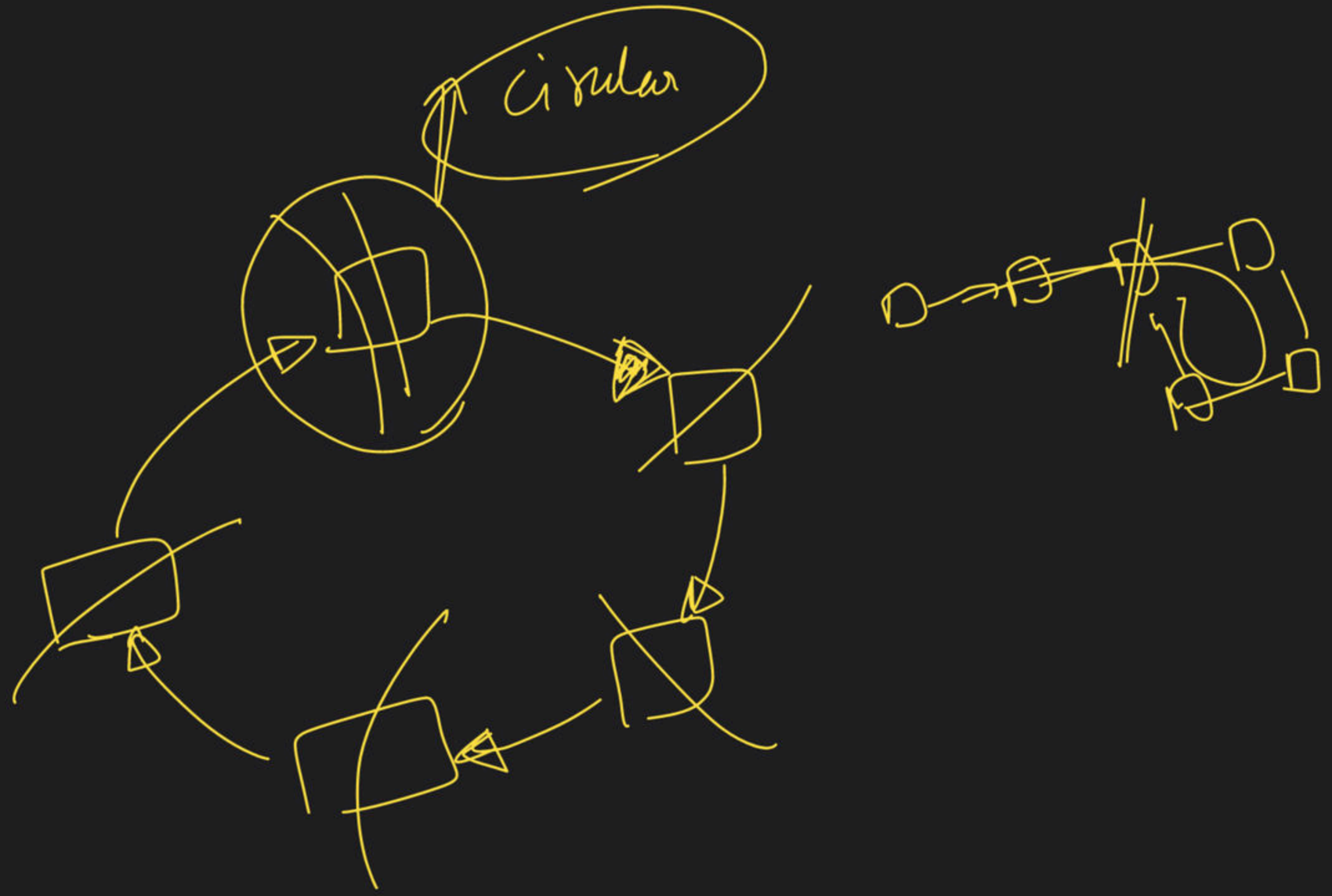

if (m.find('a') != m.end())
{
}


no of demand → n
free box → b
n / b
< 0.7
Good
Load factor
n >> k
O(K)
O(1)
babbar
K size
O(K)


i
Babbar
love
Rahul
Chutki
O(n)
O(1)
M · F
1 · F < 0 · 7

(1) i/p → Array → "Thiruvananthapuram"




circular

going Bye Bye

u all

Stay → it's Single good