Maps & Tries Class - 2

Special class

Love Babbar • Sept 5, 2024

→ Hash function

unordered_map

↳ Bucket Array

2 days

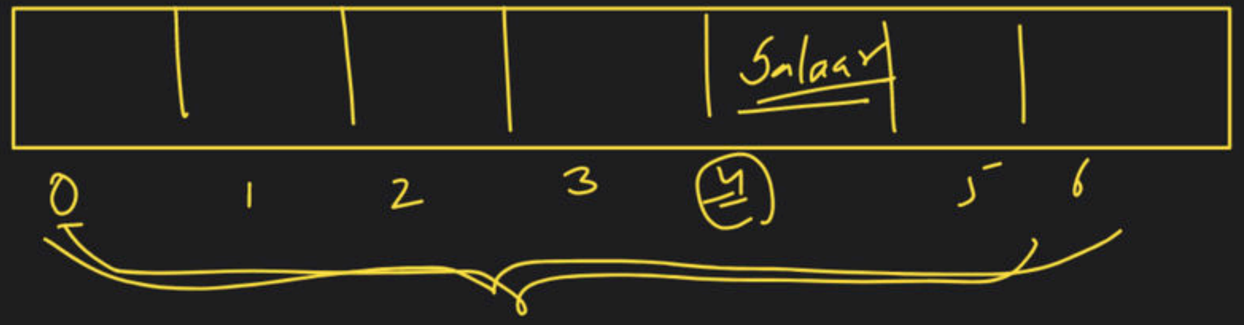
→ new link

↳ pair

"love" → "Salaar"

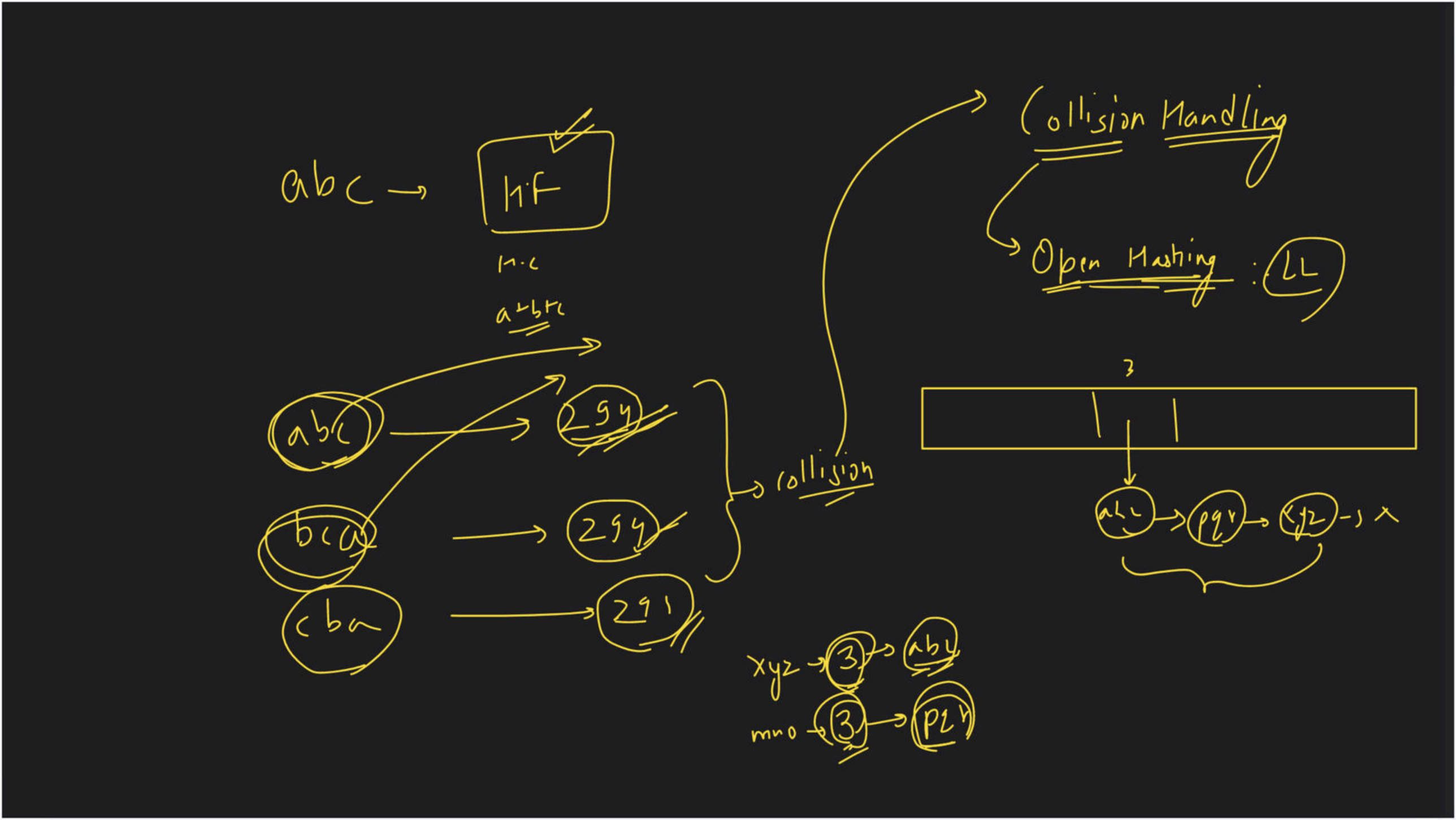
"Manish" → "KGF"

"Kunal" → "Rocky"



"love" → H.F → 4

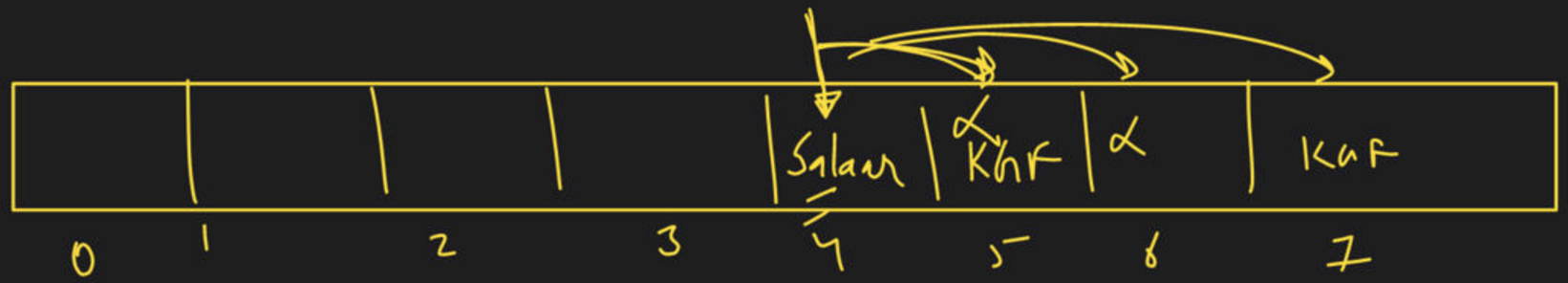
"love" → [H.F] → 241 [H.F] → 4


abc →
hf
m.c
a+b+c
abc
bca
cba
291
294
294
collision
Collision Handling
Open Hashing : LL
3
abc
pqr
xyz
Xyz
3
abc
mno
3
pqr

↳ Closed Addressing :- [Next free space]

cubic Probing

$h(i) + i^3$



Linear Probing

Love (4) → Salaar

Manik (4) → KGF

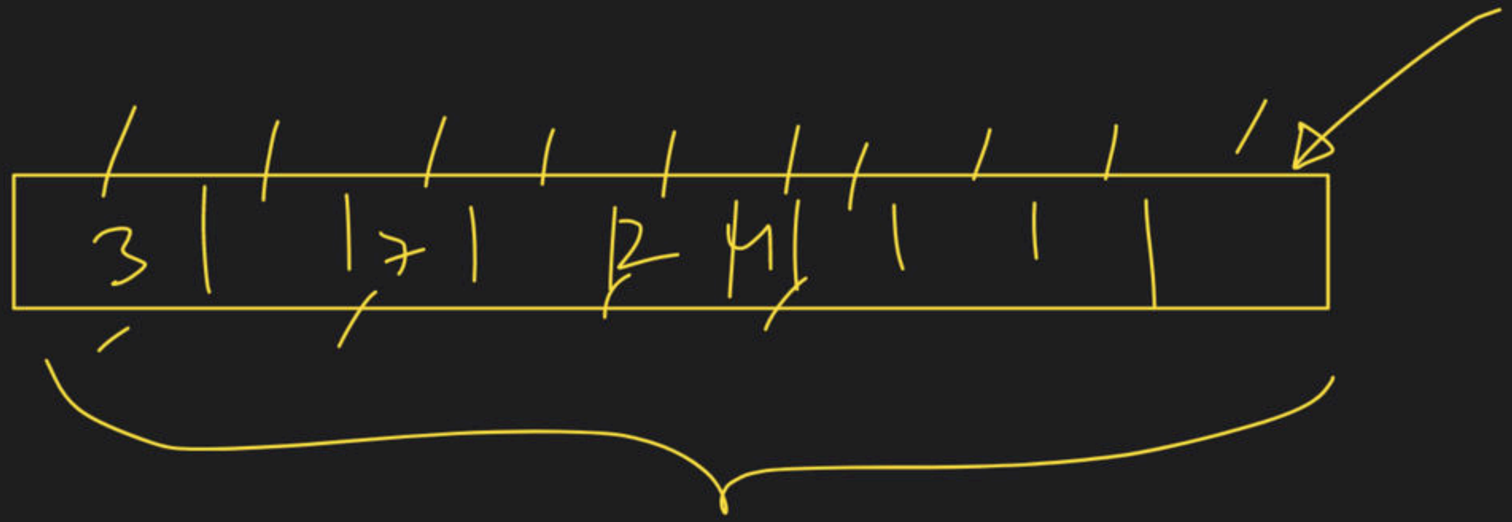
$h(i) + i$

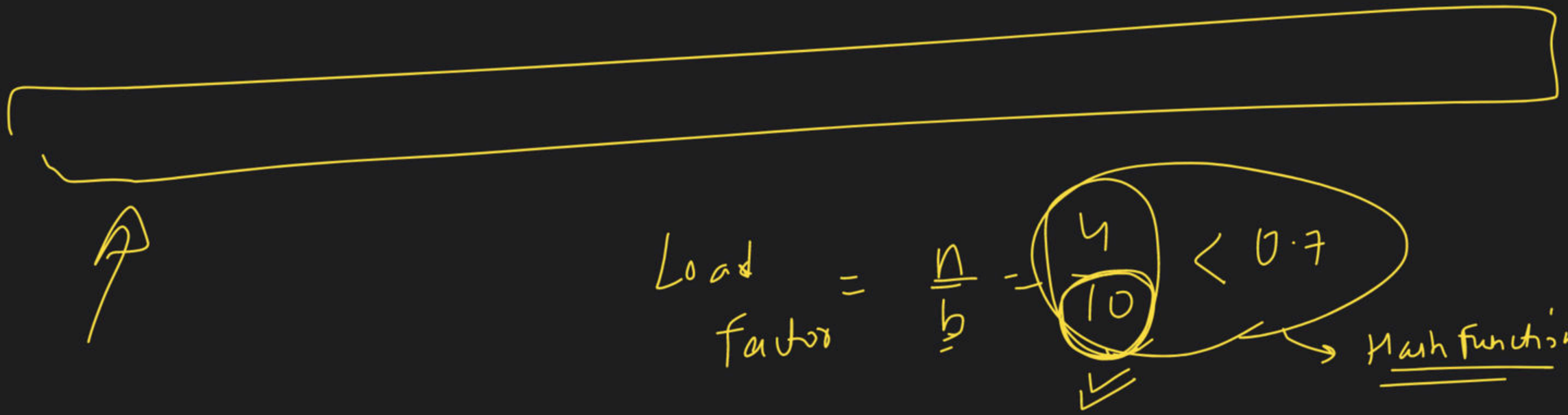
$i \to 1, 2, 3, 4, 5 \ldots$

$h(i) + i^2$

Quadratic Probing

$i \to 1, 2, 3, 4 \ldots$


3
1
1
7
1
2
4
1
1
1

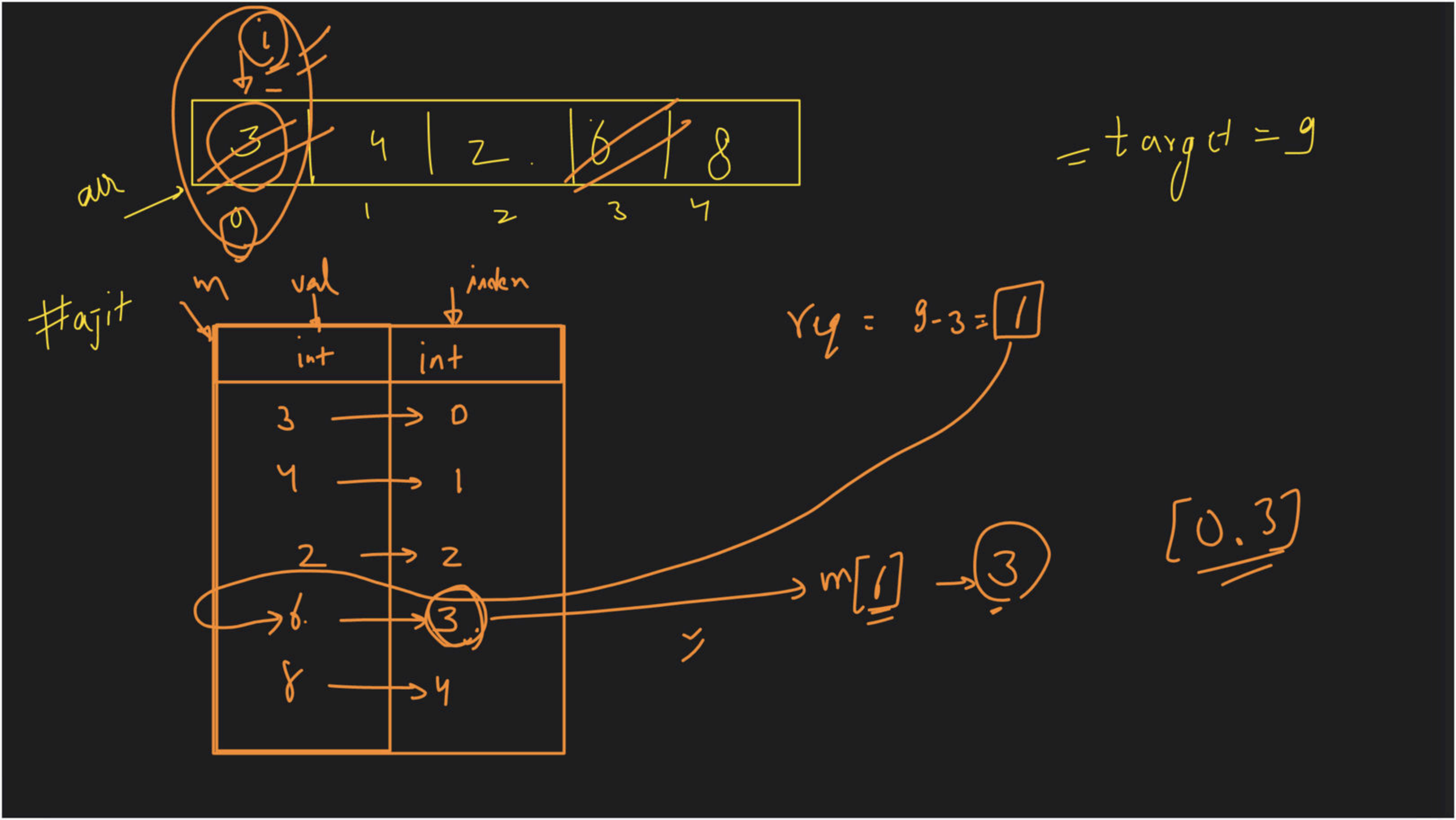

Load factor = n/b
4
10
< 0.7
Hash Function

curr + x = target

x = target - curr

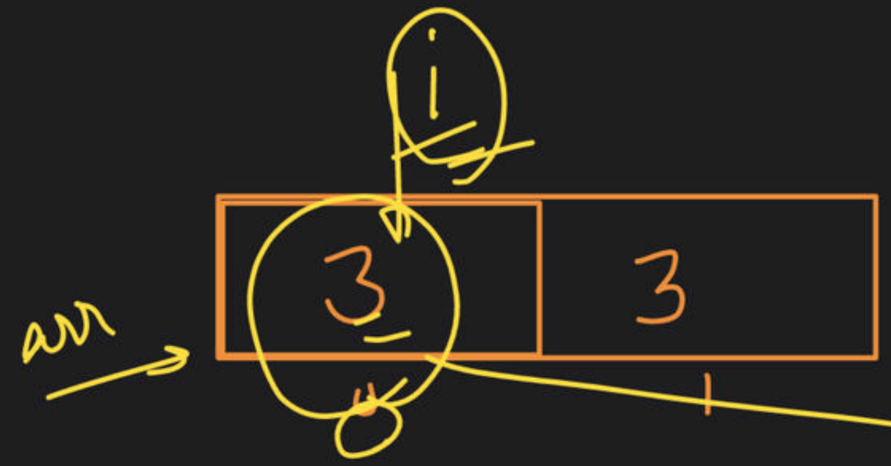
value

map - arr


arr
3
4
2
6
8
1
2
3
4
target = 9
#ajit
m
val
index
int
int
3 → 0
4 → 1
2 → 2
6 → 3
8 → 4
req = 9-3 = 1
m[1] → 3
[0,3]

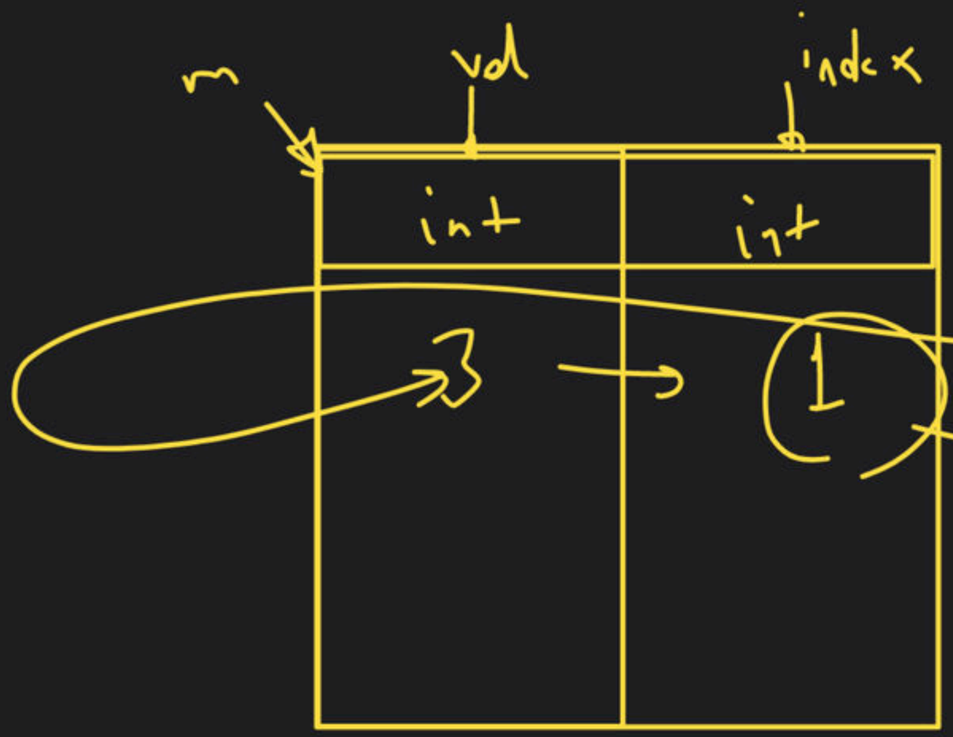
lovebabbar3@gmail.com

target = 6



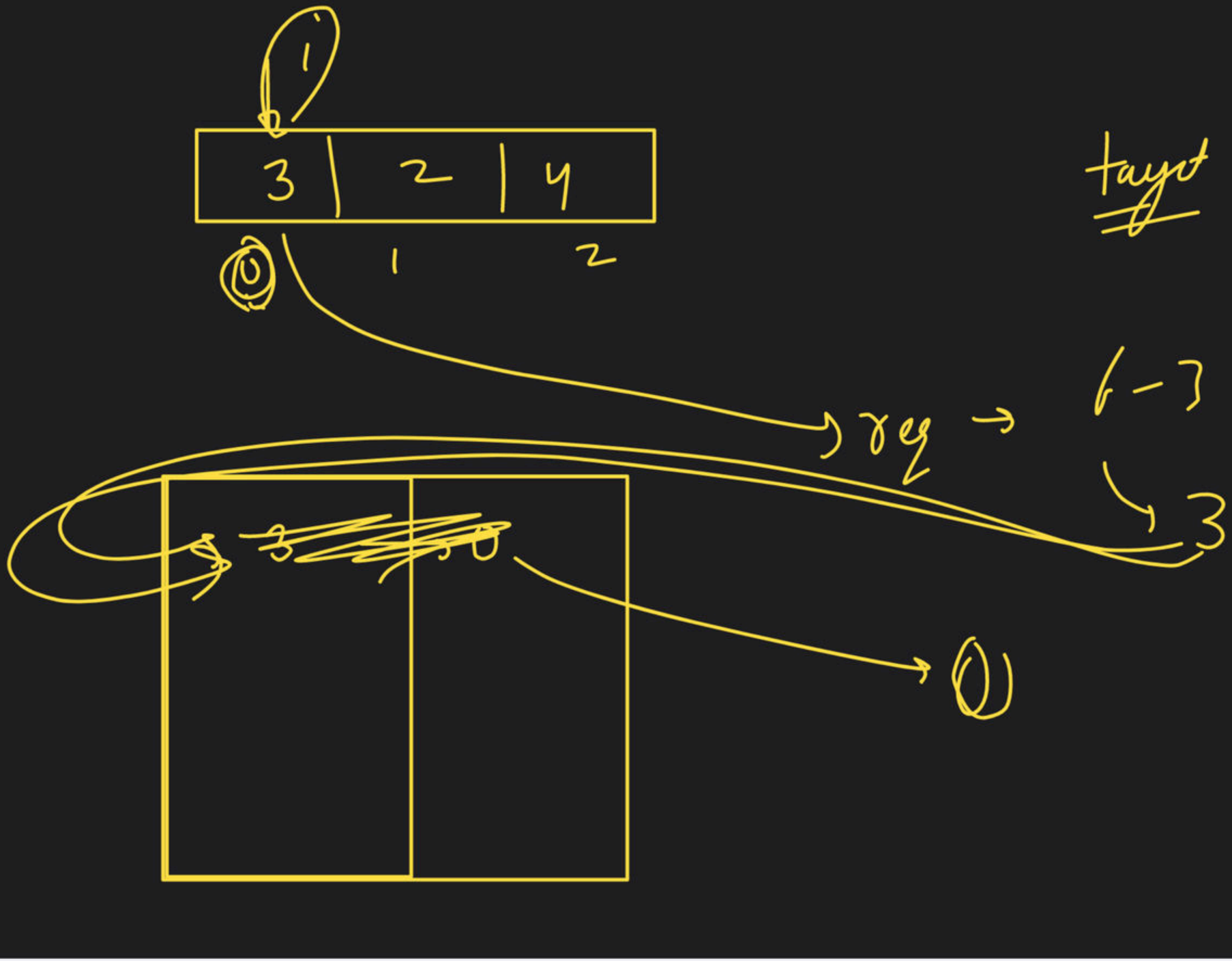
curr = 3

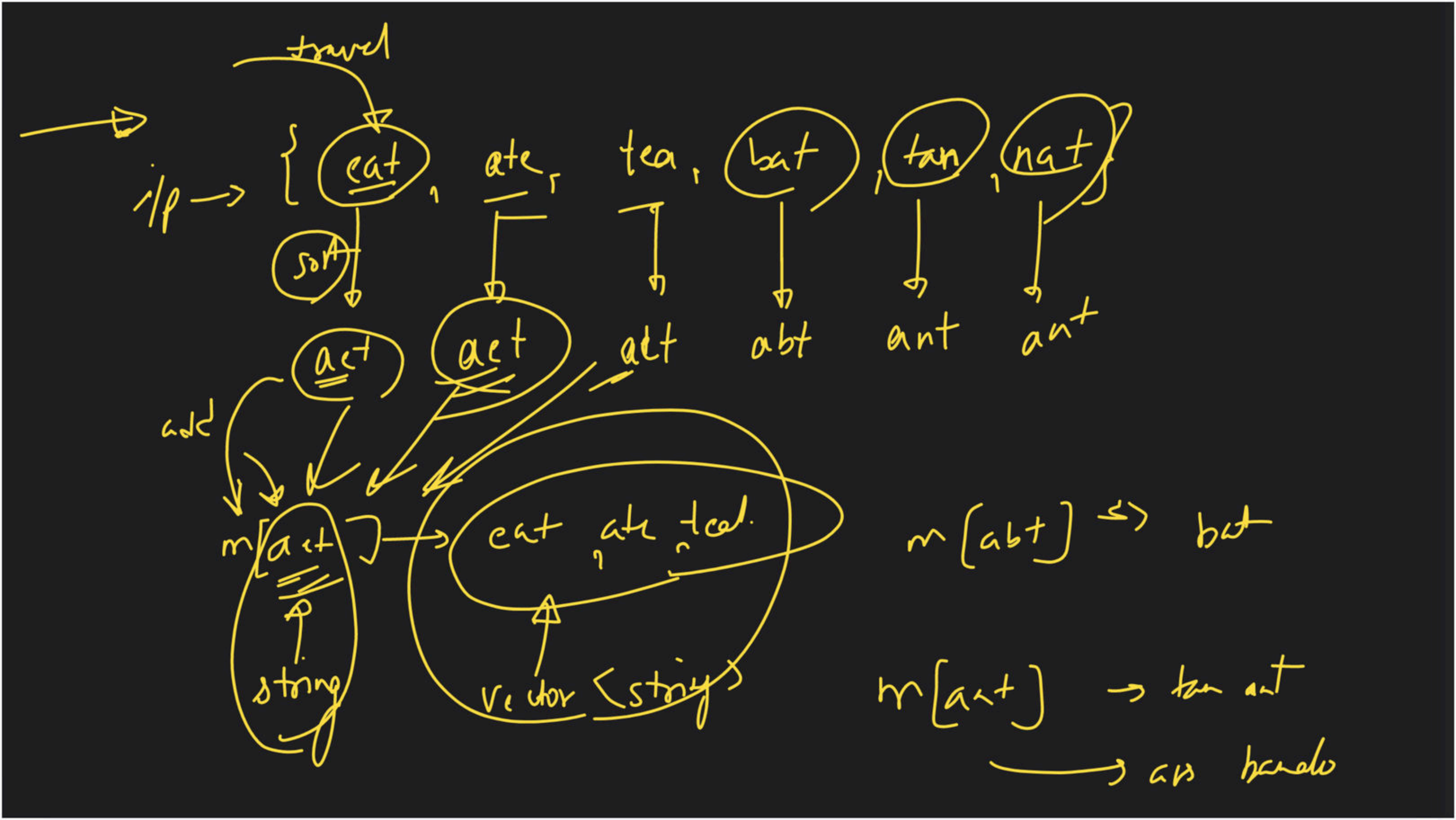
req = 6 - 3



[1, 1]

[0, 1)


3
2
4
1
2
target = 6
req → [-]
3


travel
i/p ->
cat
ate
tea
bat
tan
nat
sort
act
act
act
abt
ant
ant
add
m[act]
cat ate tea
vector<string>
string
m[abt] => bat
m[ant] -> tan ant
-> as handle

original → ~~"tan"~~ ~~ate~~ bat eat nat tea

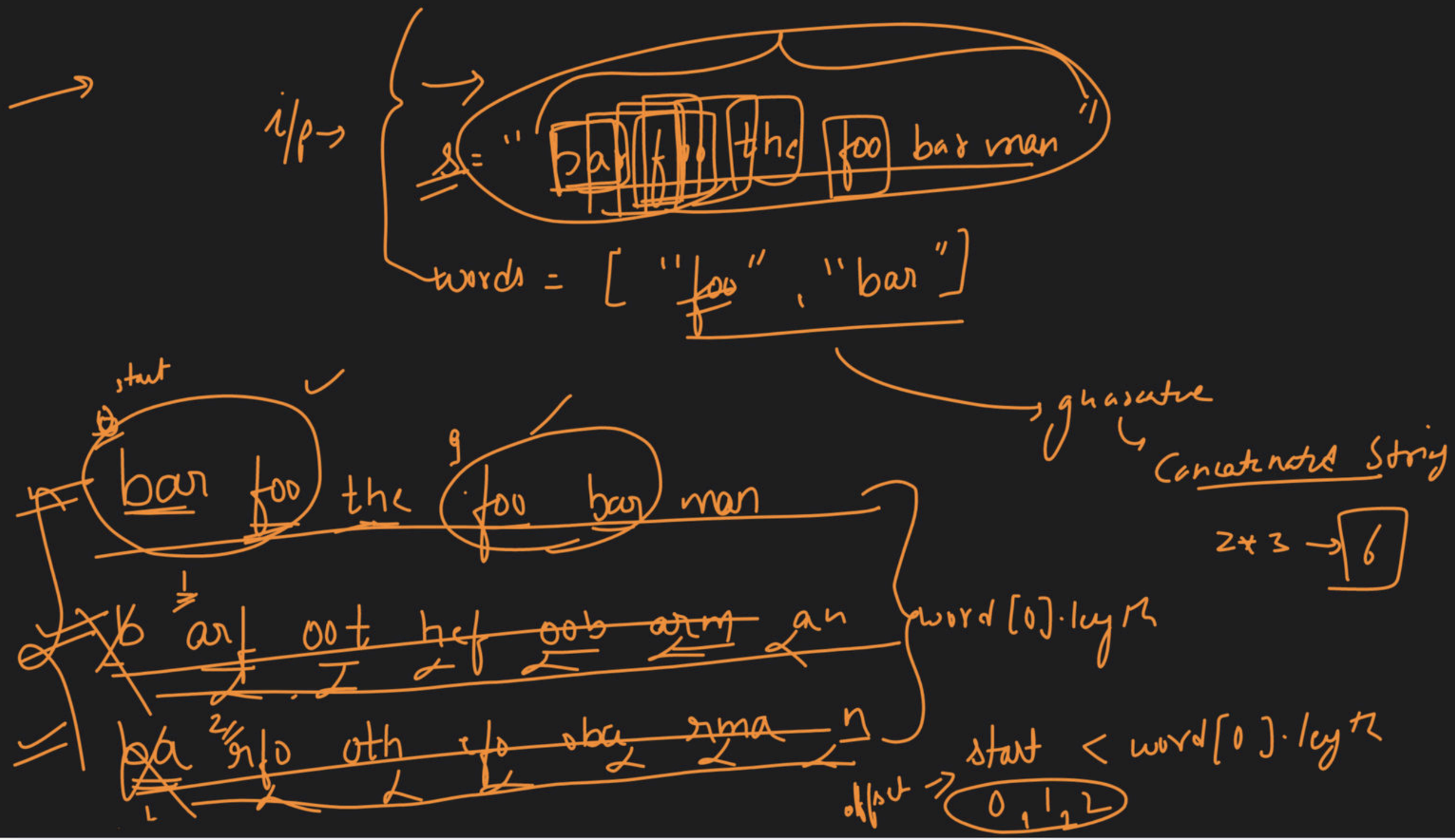
copy = ~~tea~~ ate bat eat nat tea

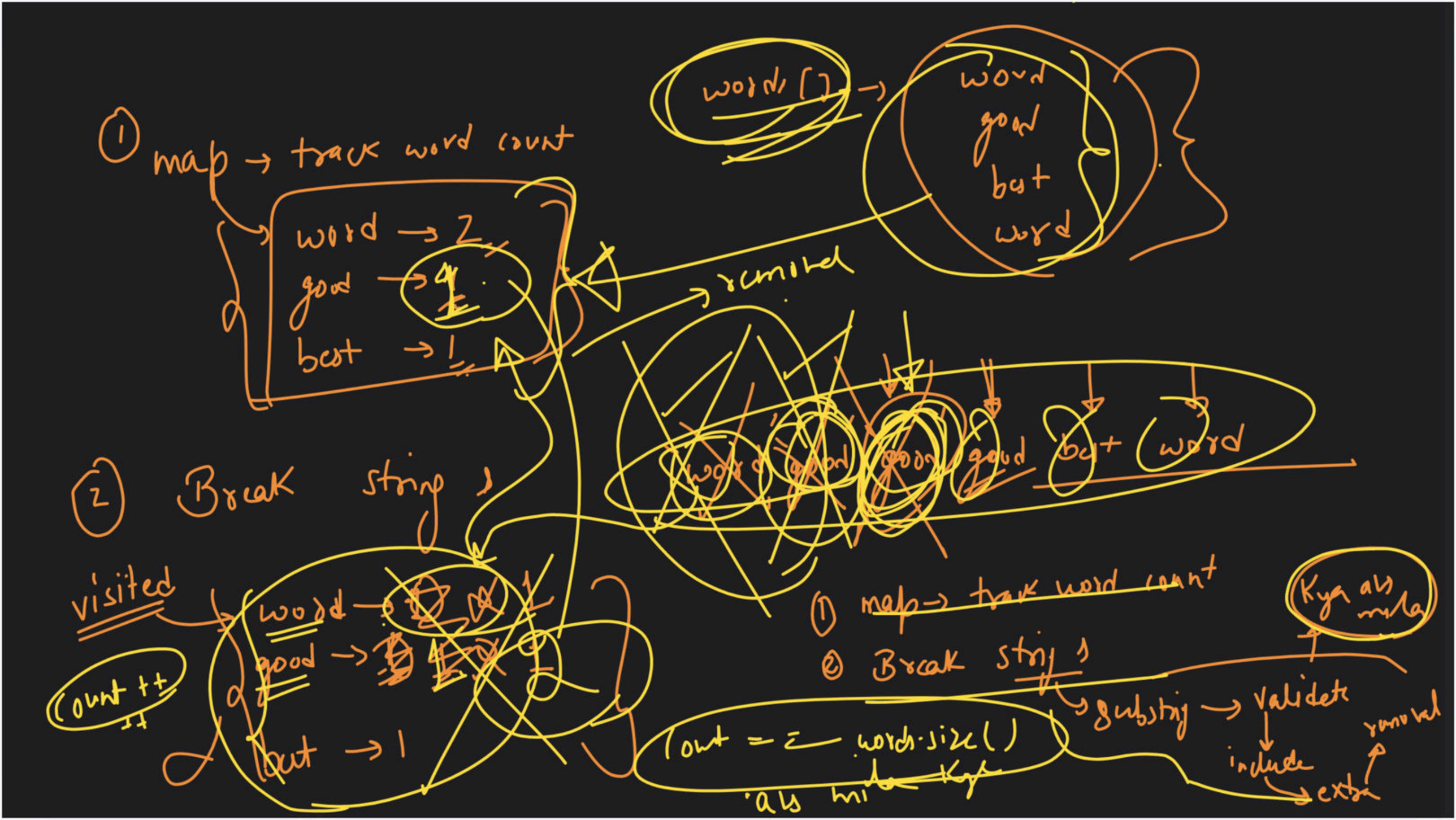
sort sort sort sort sort sort

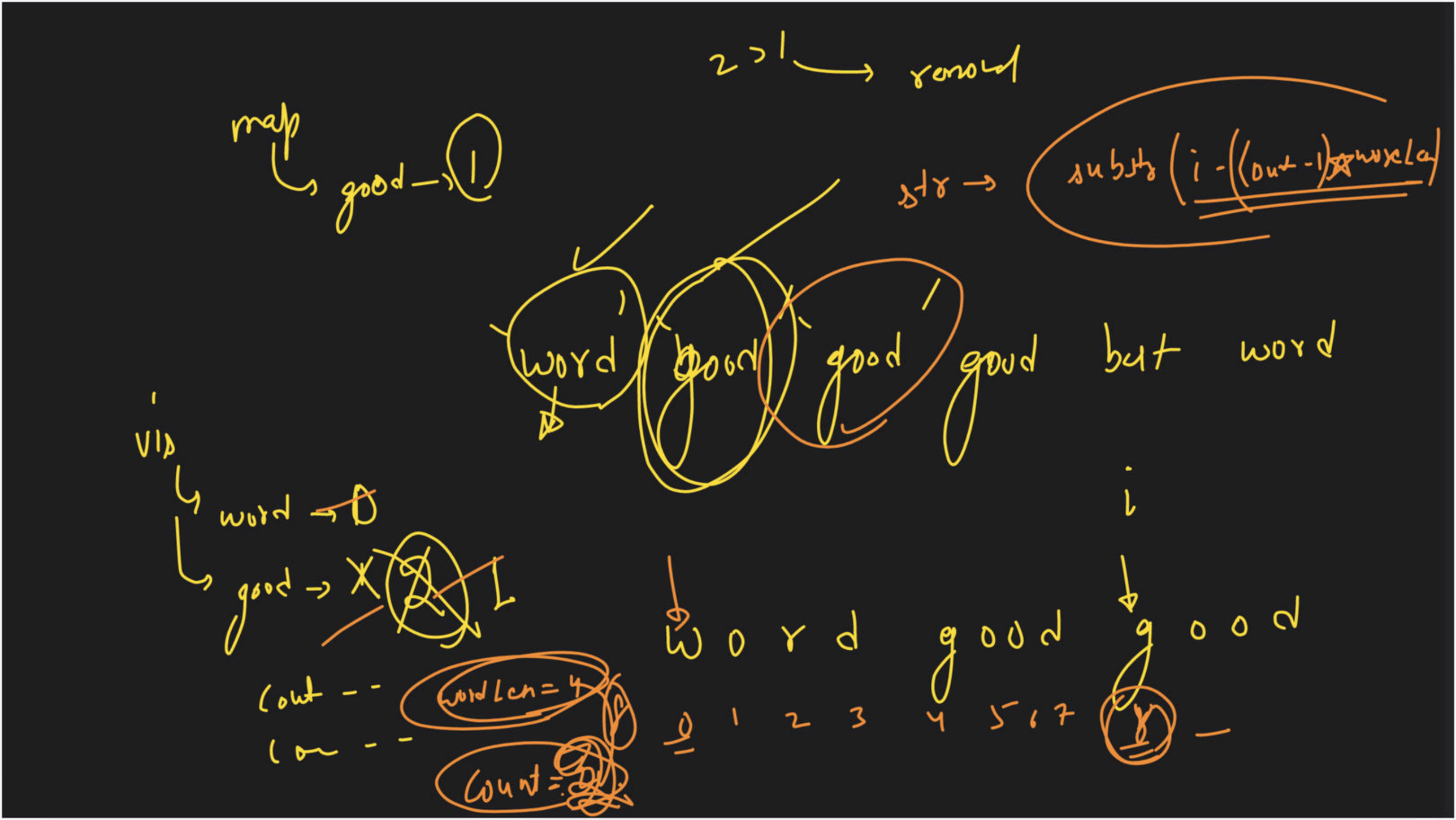
copy = ant act abt act ant act

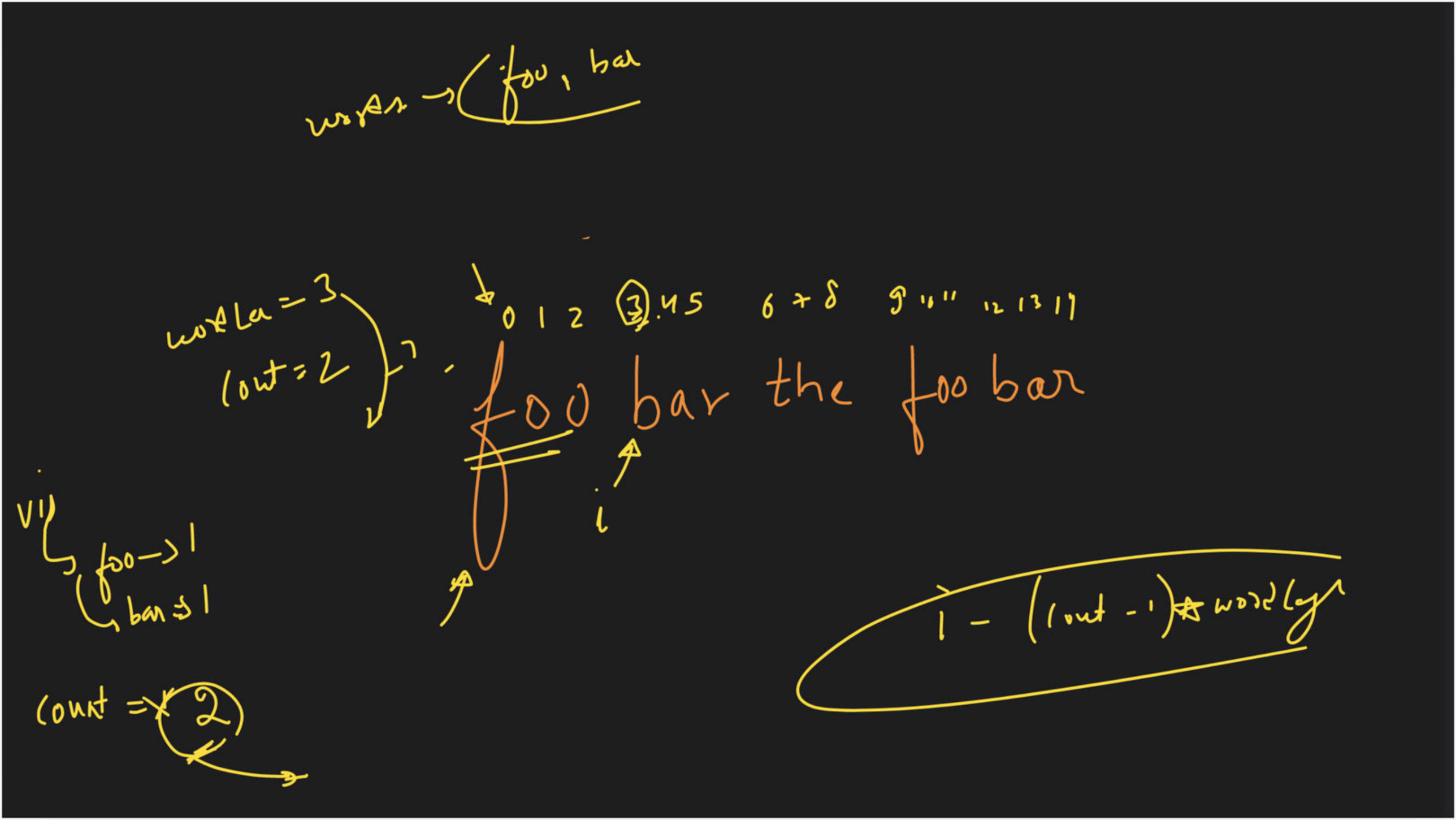
ans → { tan, nat } { ate, eat, tea } { bat }

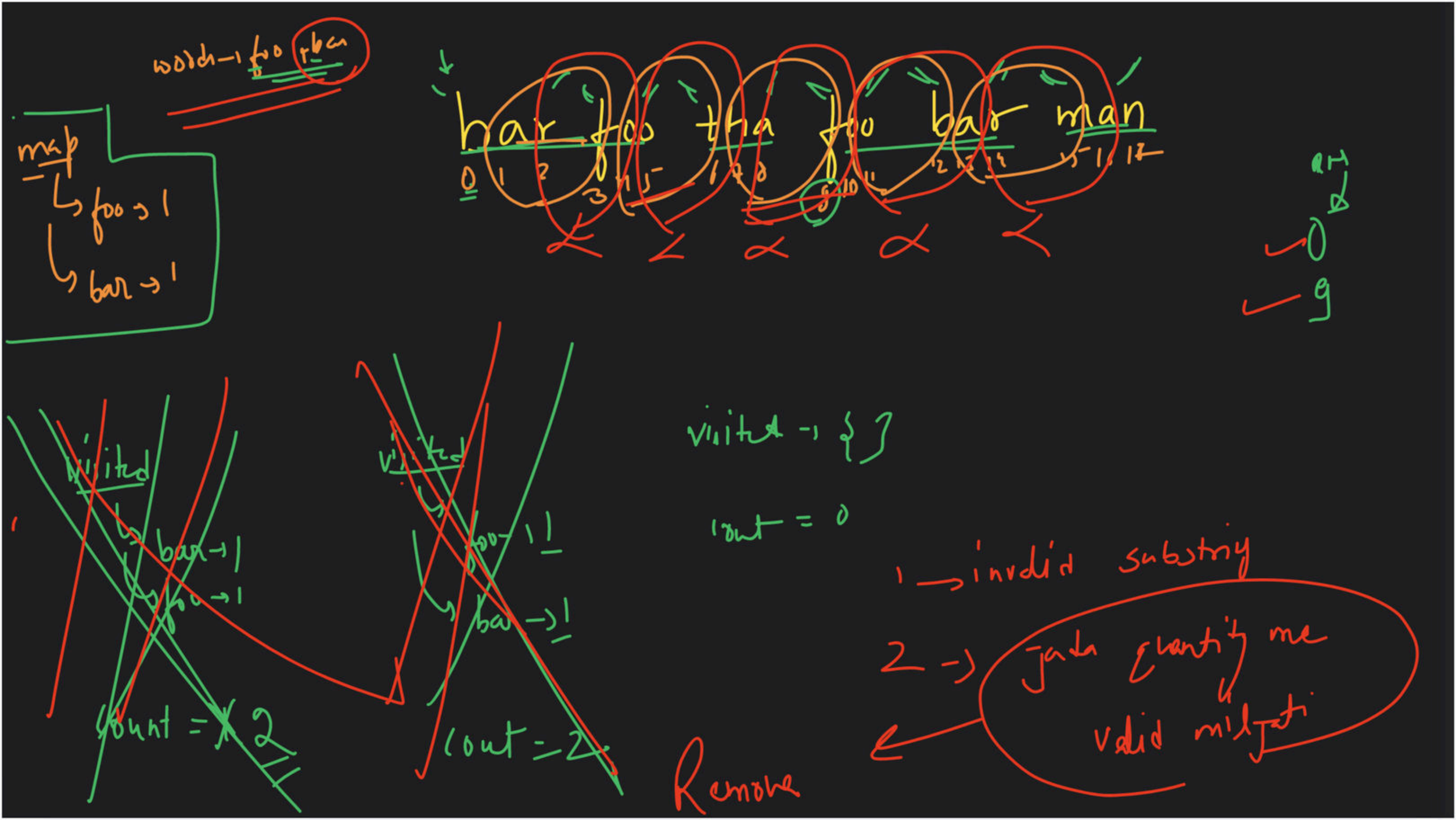
string	vector<string>
ant	tan, nat
act	ate, eat, tea
abt	bat


i/p →
s = "barfoothefoobarman"
words = ["foo", "bar"]
start
g
barfoothefoobarman
concatenated String
2 * 3 → 6
word[0].length
start < word[0].length
offset → 0, 1, 2


words[]
word
good
best
word
1 map → track word count
word → 2
good → 1
best → 1
returned
2 Break string
visited
word
good
best → 1
count ++
word good good good best word
1 map → track word count
2 Break string
substring → validate
include
extra
removed
count = words.size()
Kya ans milega


2 > 1 → removed
map
good → 1
str →
substr(i - ((count - 1) * wordLen))
word good good good but word
i
VIP
word → 0
good → X
count--
wordLen = 4
count = 0
word good good
0 1 2 3 4 5 6 7
i
_


words → foo, bar
wordLen = 3
count = 2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
foo bar the foo bar
i
VI
foo → 1
bar → 1
count = 2
i - (count - 1) * wordLen


word-1 foo bar
map
foo → 1
bar → 1
bar foo tha foo bar man
visited
bar → 1
foo → 1
count = 2
visited
foo → 1
bar → 1
count = 2
visited → { }
count = 0
1 → invalid substring
2 → jaha equality me
valid milgaya
Remove