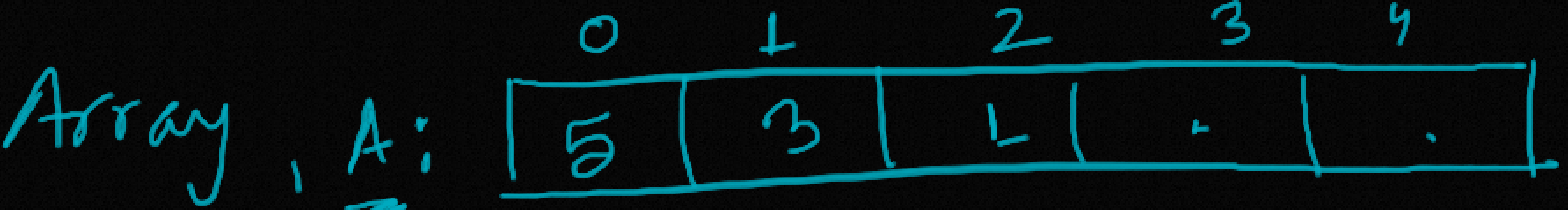
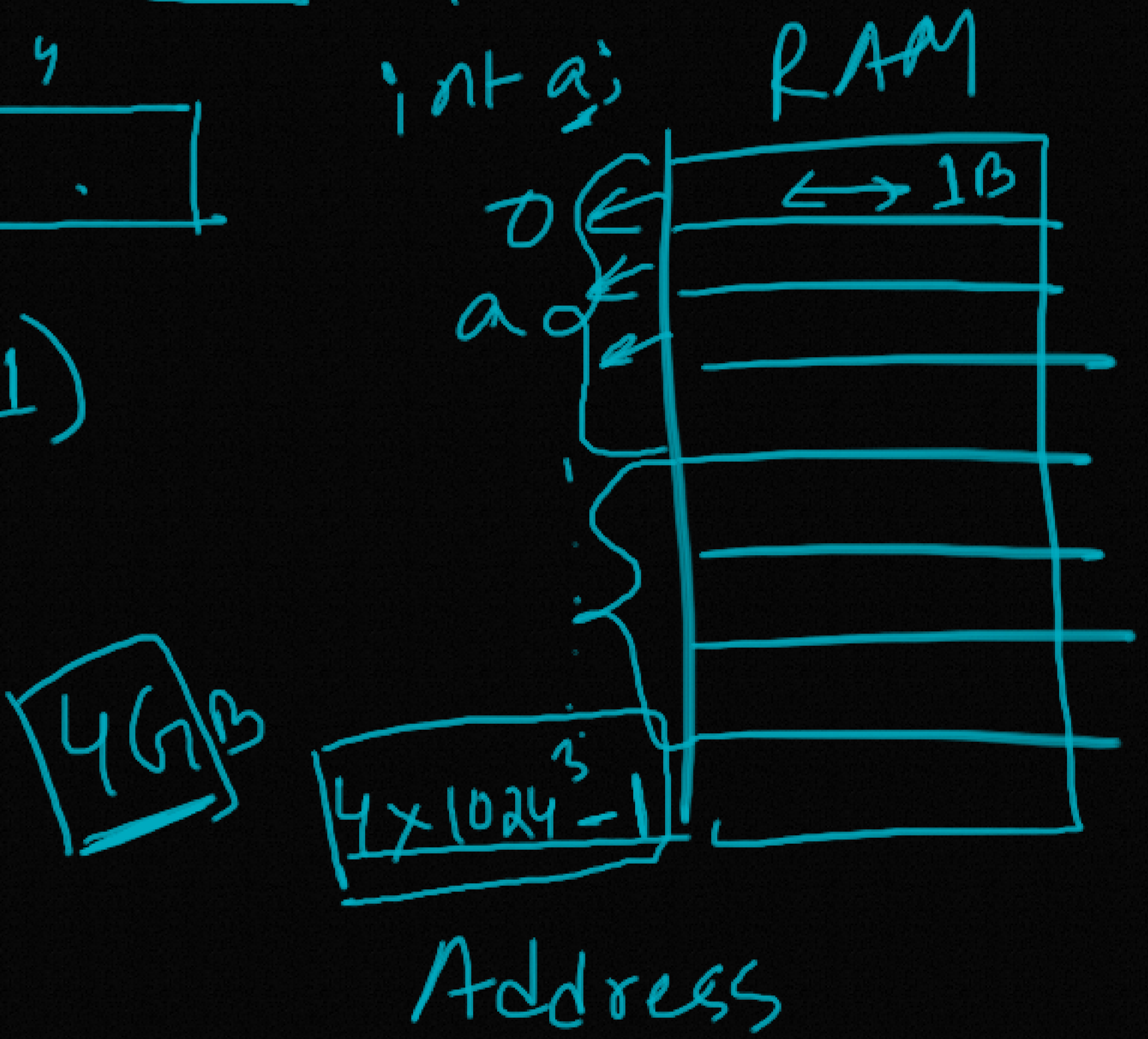
Data Structures



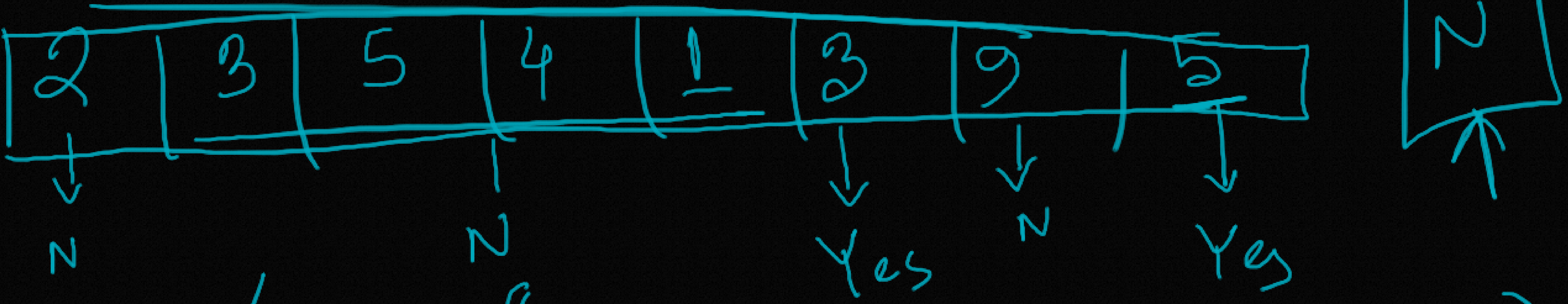
$A[0]$ $\quad A[K] \rightarrow O(1)$

int → 4B

int a;



Sequence of positive integers



$\underline{10}$ / 1 $-$ $\underline{10^8}$

Naive

✓ 1st $\rightarrow$ 0

✓ 2nd $\rightarrow$ $\underline{1}$

3rd $\rightarrow$ 2

⋮

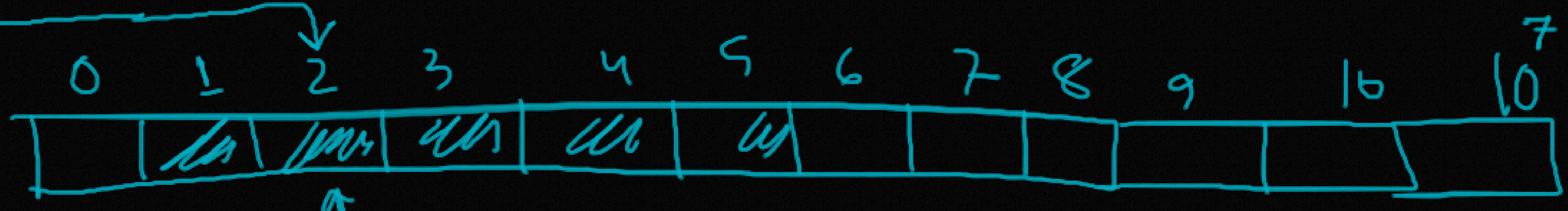
Nth $\rightarrow$ N-1 /

$$O\left(\frac{N(N-1)}{2}\right) \Rightarrow \underline{O(N^2)}$$

$$\Downarrow$$

$$O(N)$$

$\boxed{1-10}$



2 — 1

3 — 1

4 — 1

5 — 1

1

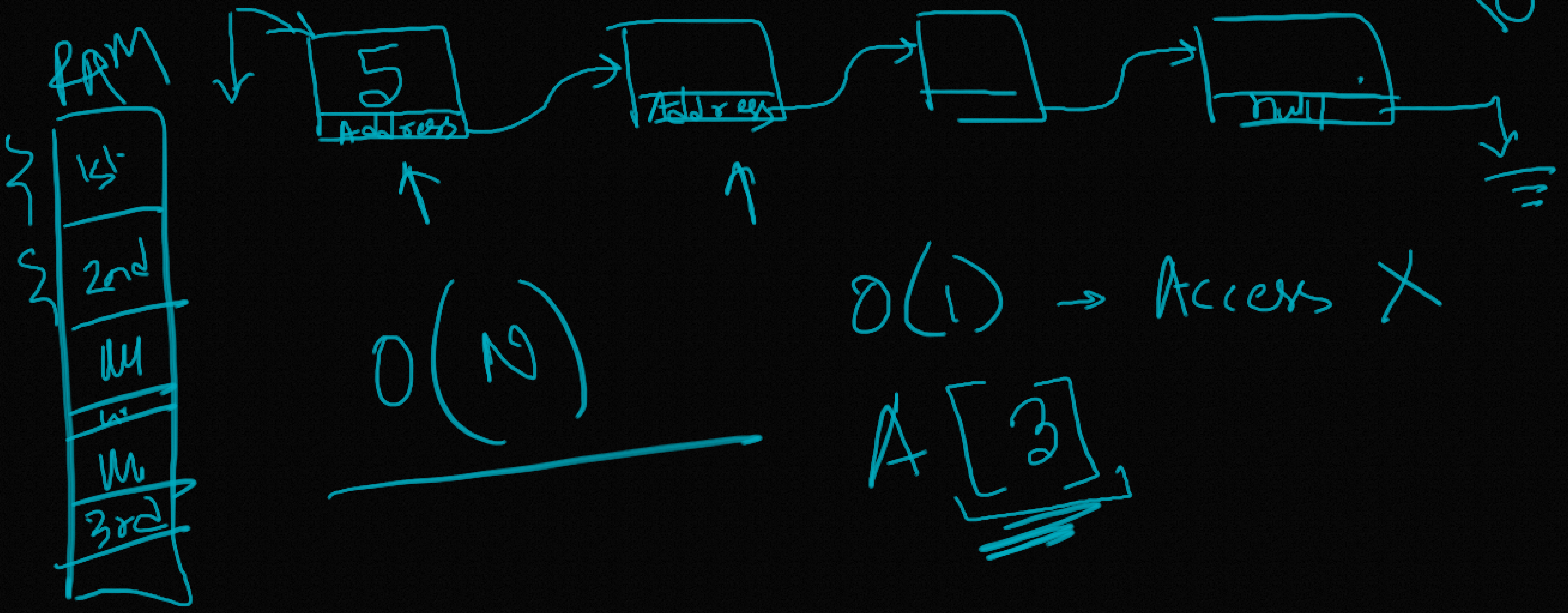
2

$1 - K$

```
A[11] = {0}
while (input, n) {
    x = A[n]
    if (x == 0) A[n] = 1    print(No)
    else print(yes)
}
```

$O(N)$

linked_list



$O(N)$

$O(1) \rightarrow$ Access ✗

A[3]

$$\begin{array}{ccccccccc} & & \downarrow & & \downarrow & & & & \\ 2 & 3 & \underline{1} & 4 & 6 & \underline{1} & 4 & 3 & 3 \end{array}$$

Max - Min = K

$\underline{1}$	2
2	$\underline{1}$
3	3
4	2
5	0
6	$\underline{1}$

Naive → $O(\underline{N^2})$

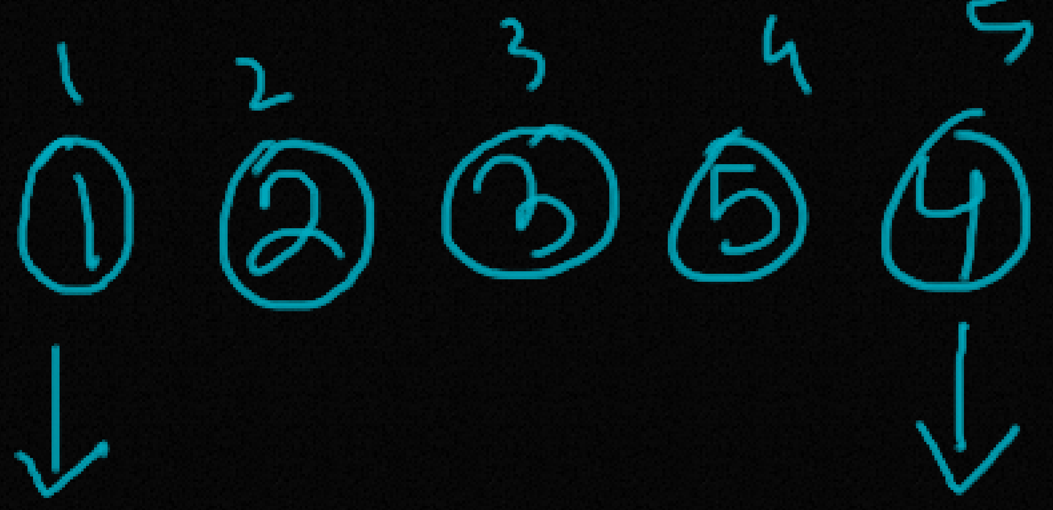
$O(\underline{K \times N})$

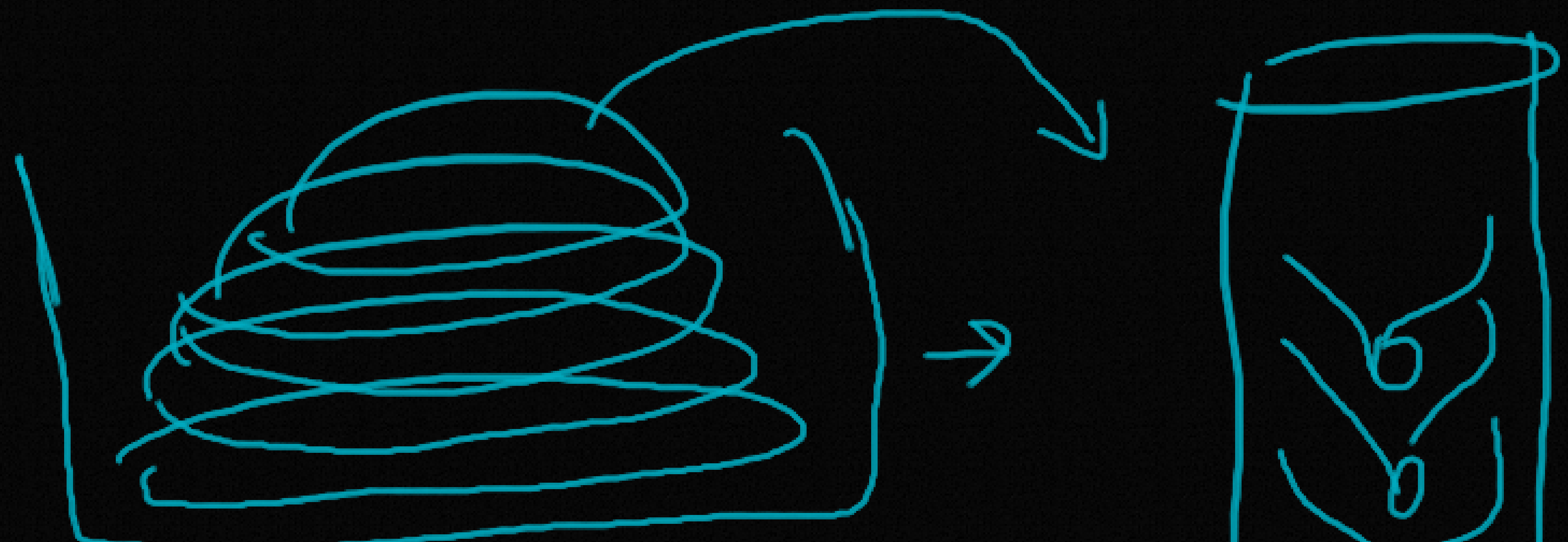
$O(\underline{K})$

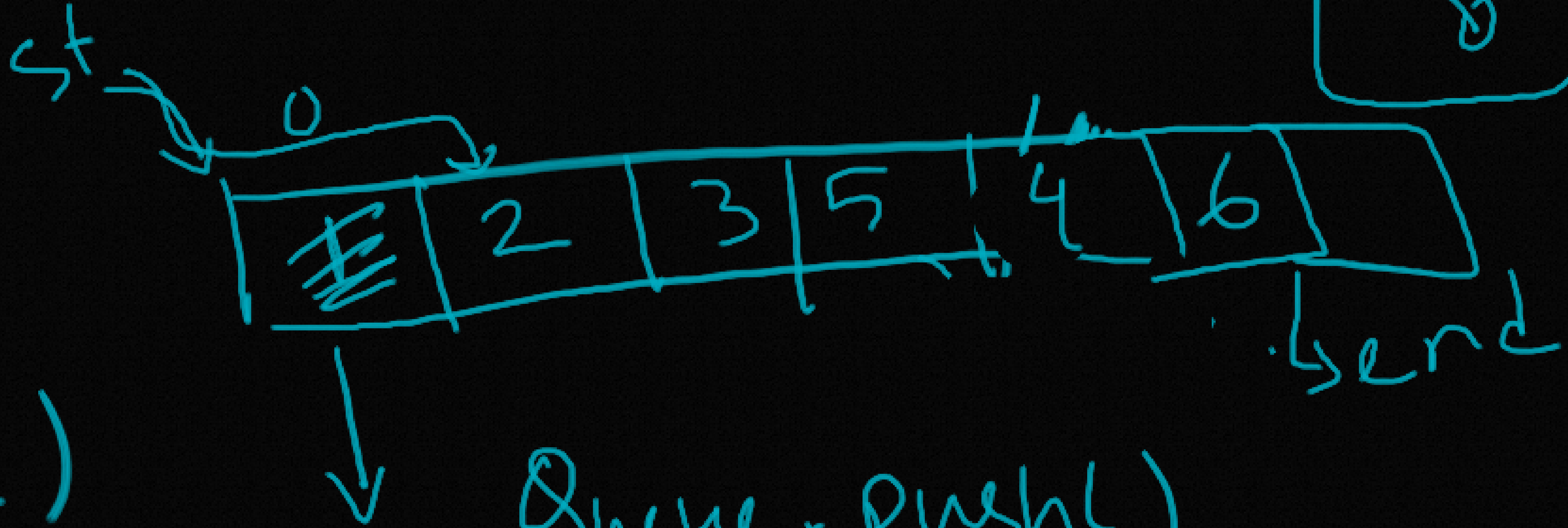
$\textcircled{1}$	2	3	4	5	6
2	~~2~~1				

Stack, Queue, Map, Double ended Queue

$$\begin{tikzcd} \text{Stack} \arrow[d] & \text{Queue} \arrow[d] \\ \text{LIFO} & \text{FIFO} \end{tikzcd}$$







Stack.push()
Stack.pop()

Queue.push()
" .pop()

(()) → ✓

()() → ✓

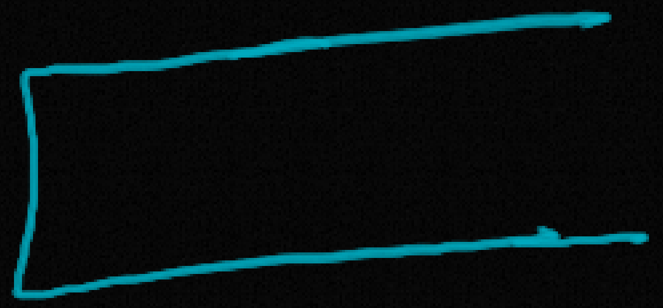
((()))() → ✓

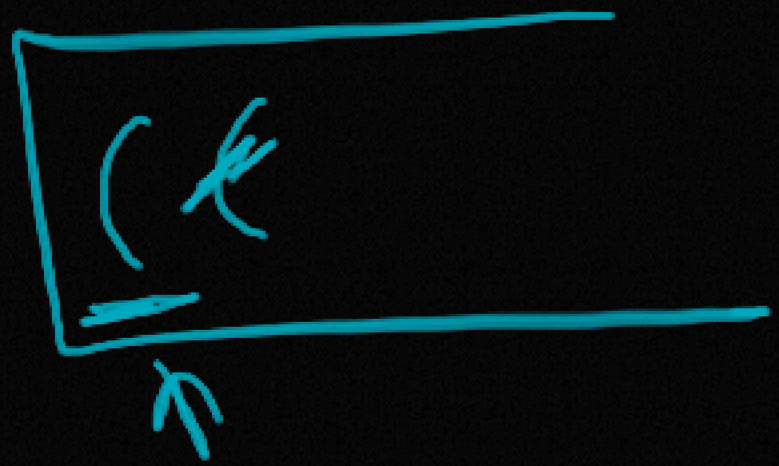
)(→ ✗ →

((→ ✗

(() → ✗ →

LIFO → Stack





$\underline{()[]} \longrightarrow$ ✓

$([]) \longrightarrow$ ✓

$(] \longrightarrow \times$

$([)] \longrightarrow \times$

$(()) \longrightarrow \underline{4}$

$(\underline{()} \longrightarrow 2$

$)\underline{()()} \longrightarrow 4$

Double Ended Queue

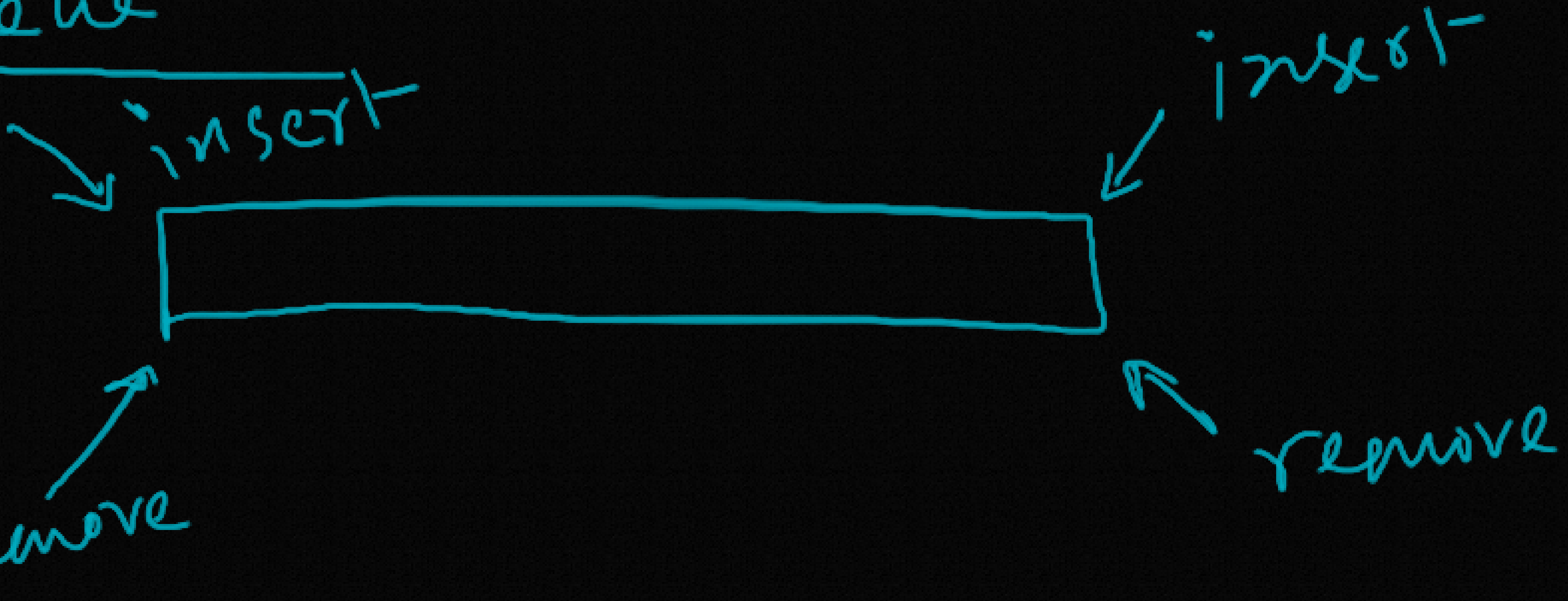
Q: FIFO

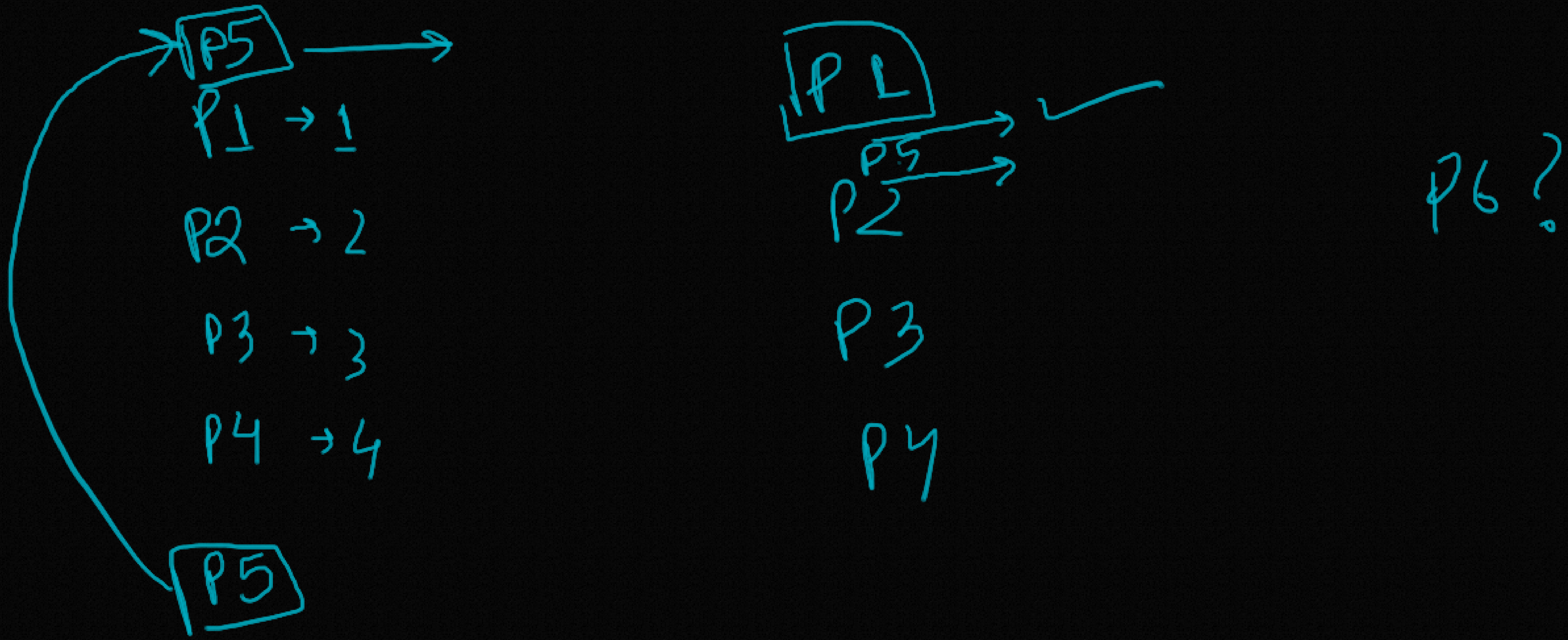
Queue.push()

Queue.pop()

D_Q.~~push~~ pop _back()

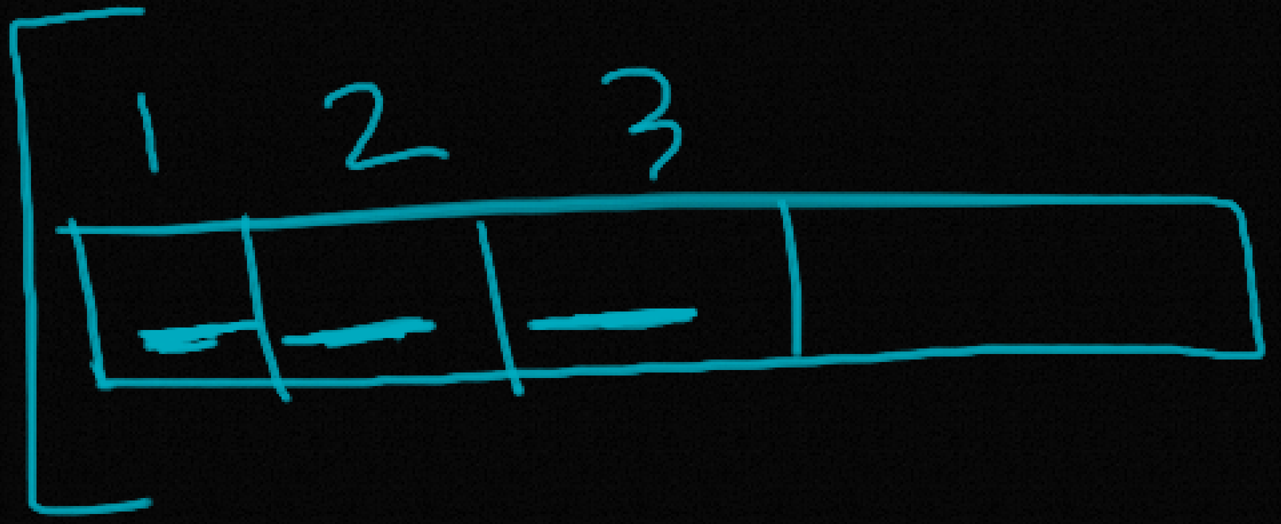
D_Q.~~push~~ pop _front()

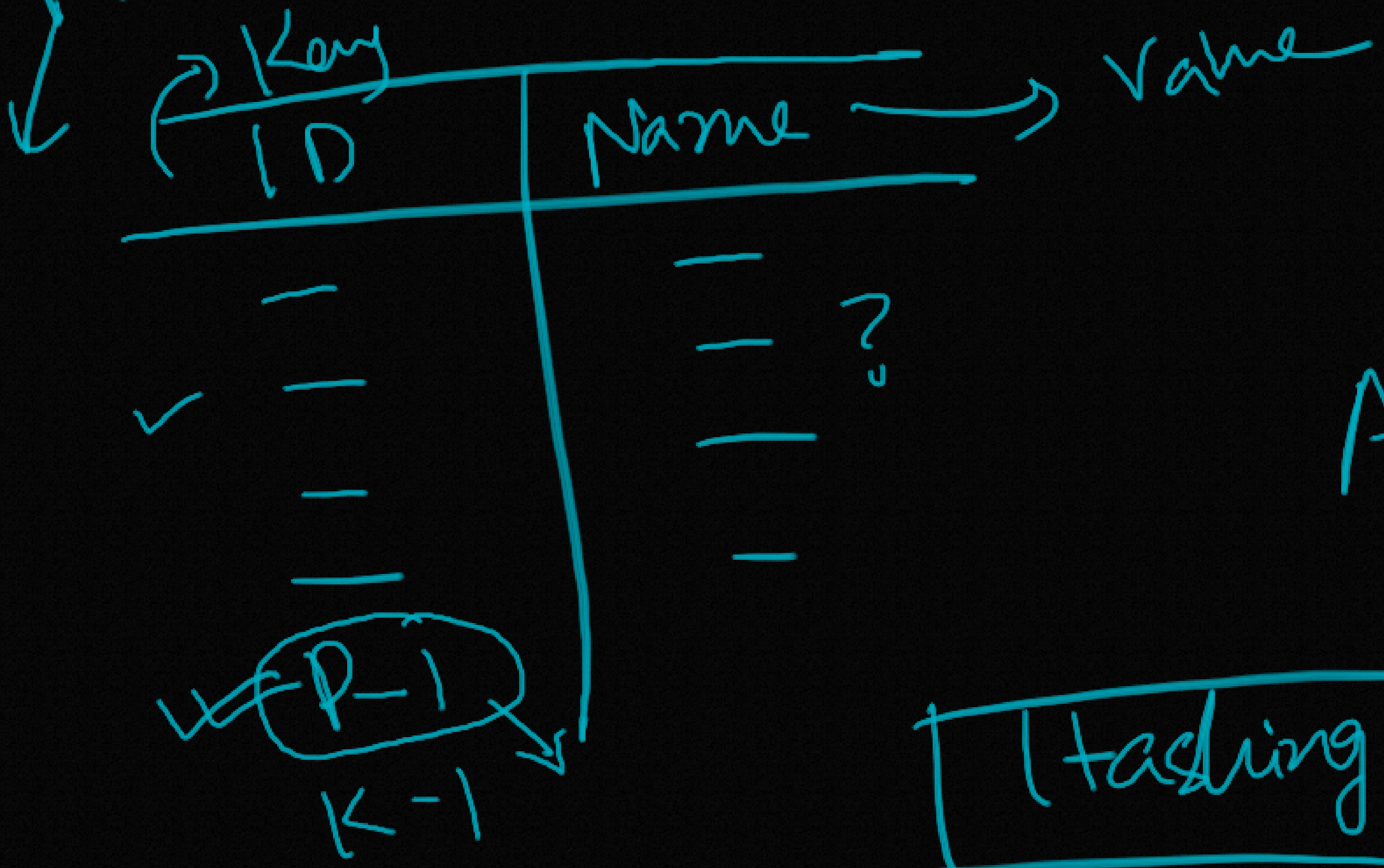



P5
P1 → 1
P2 → 2
P3 → 3
P4 → 4
P5
P1
P5
P2
P3
P4
P6 ?

MAP

(key, value)





A ["up-1"] =

Hashing