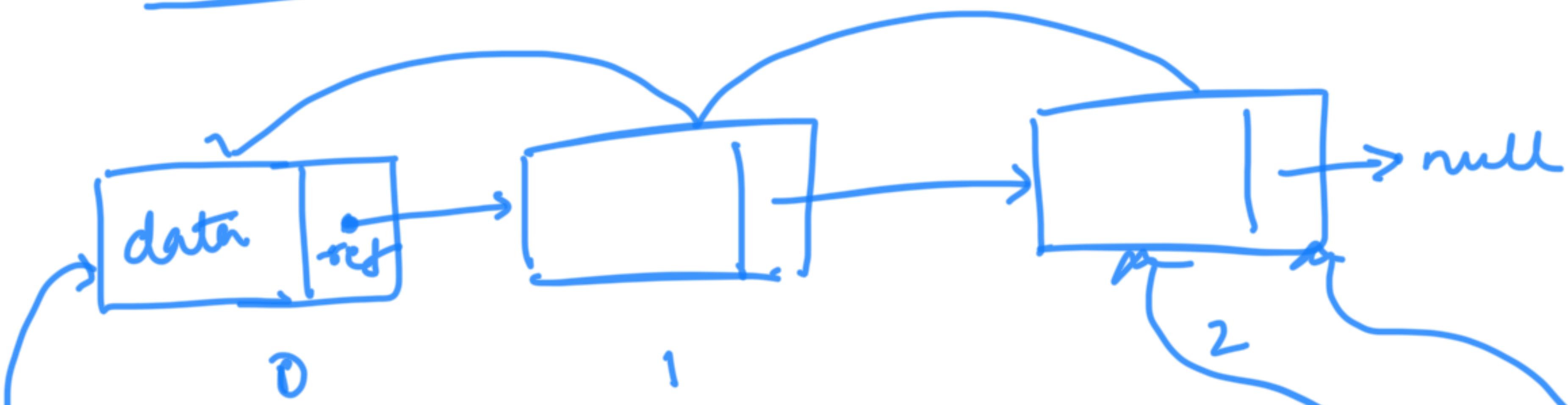
Linked list



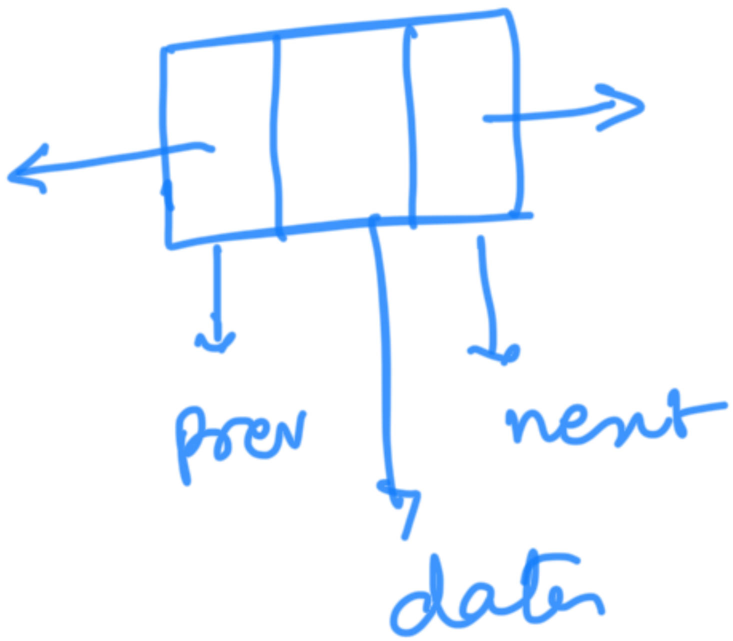
list
head

Node → LinkedListNode

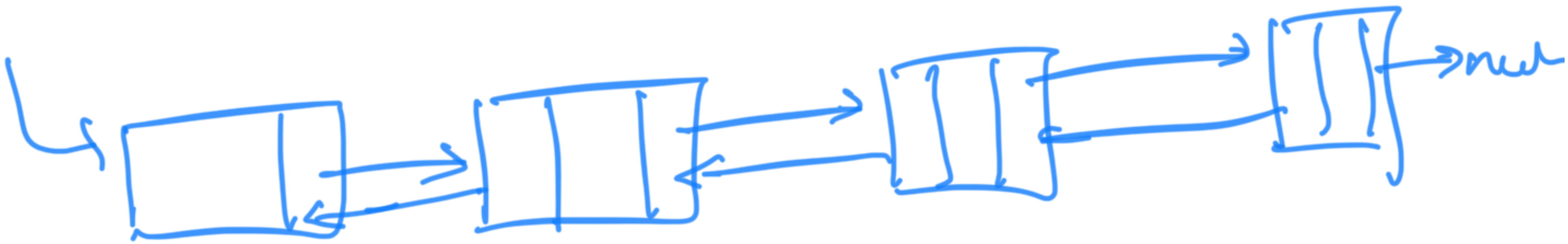
```
class {
    int val;

class Node {
    int val;
    Node next;
```

}




```
class Node {
    int val;
    Node prev
    Node next;
}
```



Doubly linked list

linked list → queue

~~Queue~~

Queue q = new linkedlist();

head
~~tail~~

~~tail~~

tail

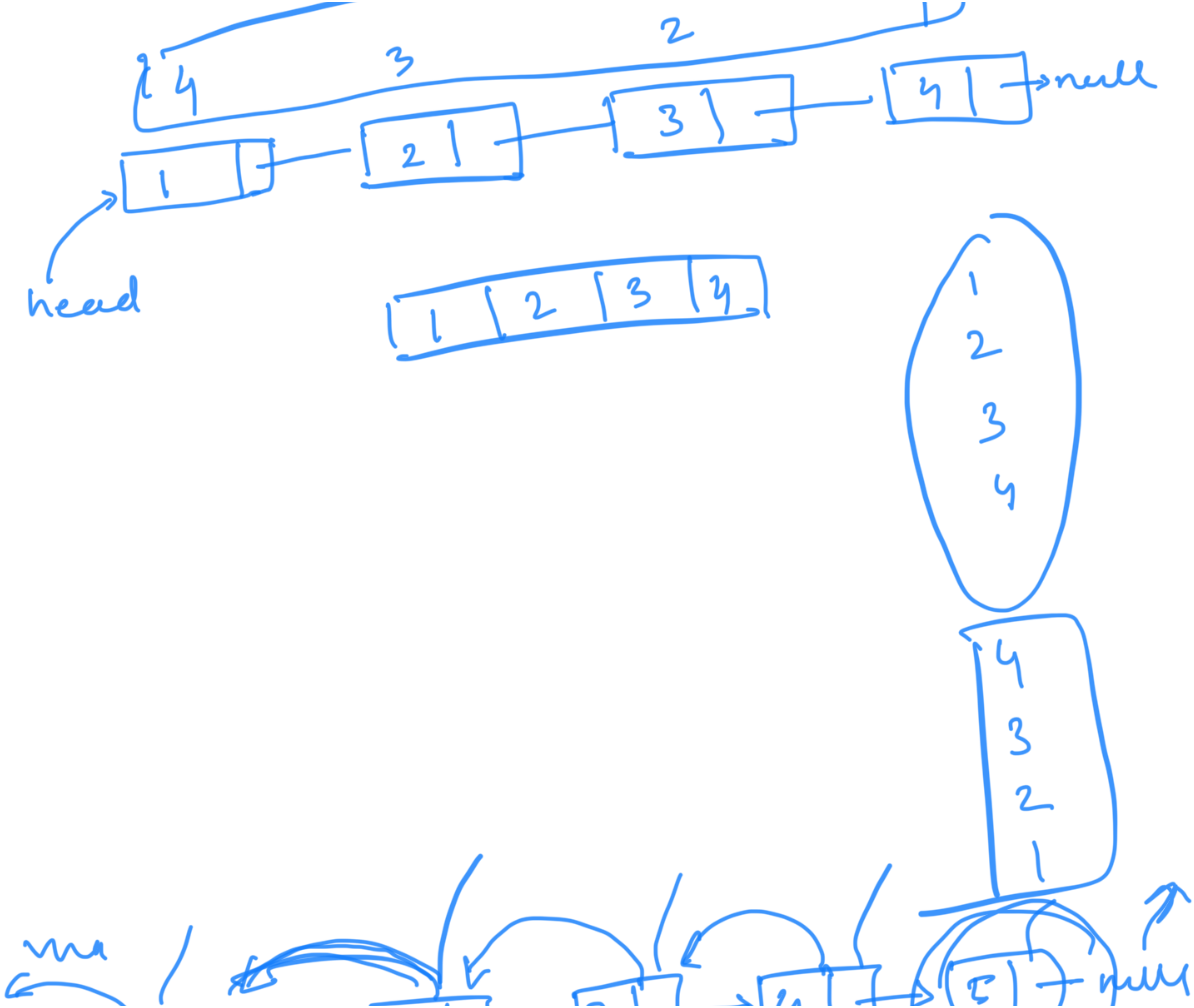
reverse a linked list

reverse in size of k

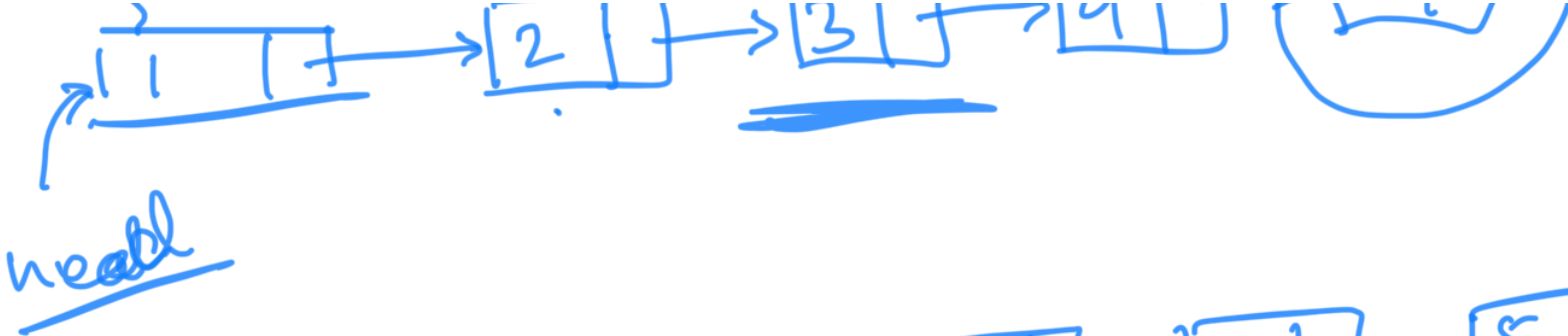
1 — 2 — 3 — 5 — 7 — 8 — 9 — 15

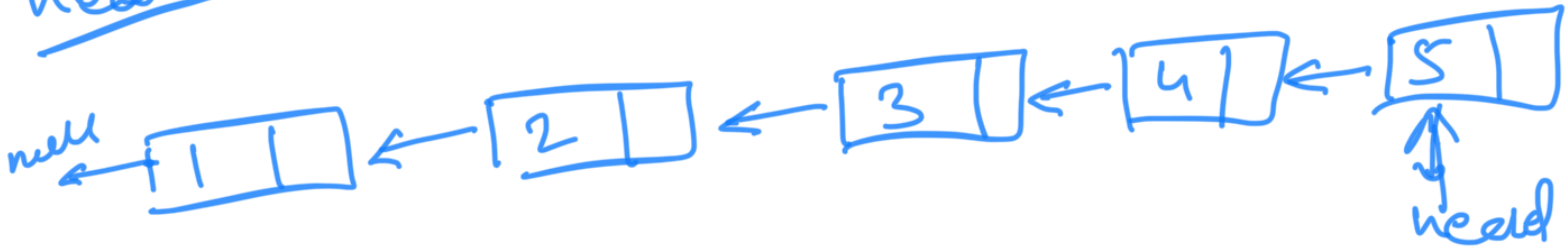
reverse in size of 3

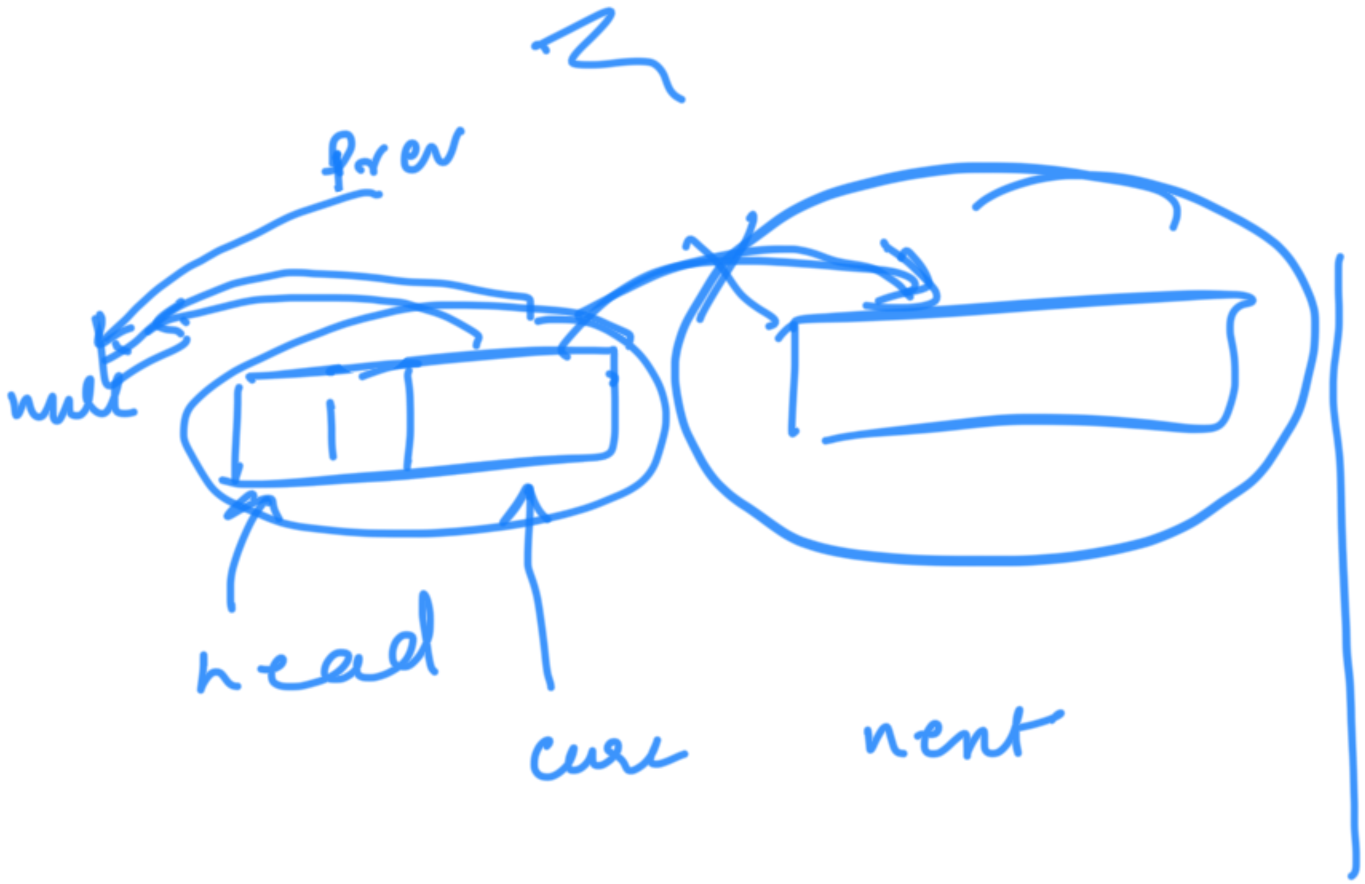
3 — 2 — 1 — 8 — 7 — 5 — 15 — 9



4
3
2
1
2
3
4
null
head
1
2
3
4
1
2
3
4
4
3
2
1
null







next = cur.next;

cur.next = prev

prev = cur

cur = next

for next iteration

```
Node reverse ( Node head )
{
    Node prev = null ;
    Node cur = head ;
    Node next = null ;

    while ( cur != null )
    {
        next = cur.next ;
        cur.next = prev ;
        prev = cur ;
        cur = next ;
    }
```

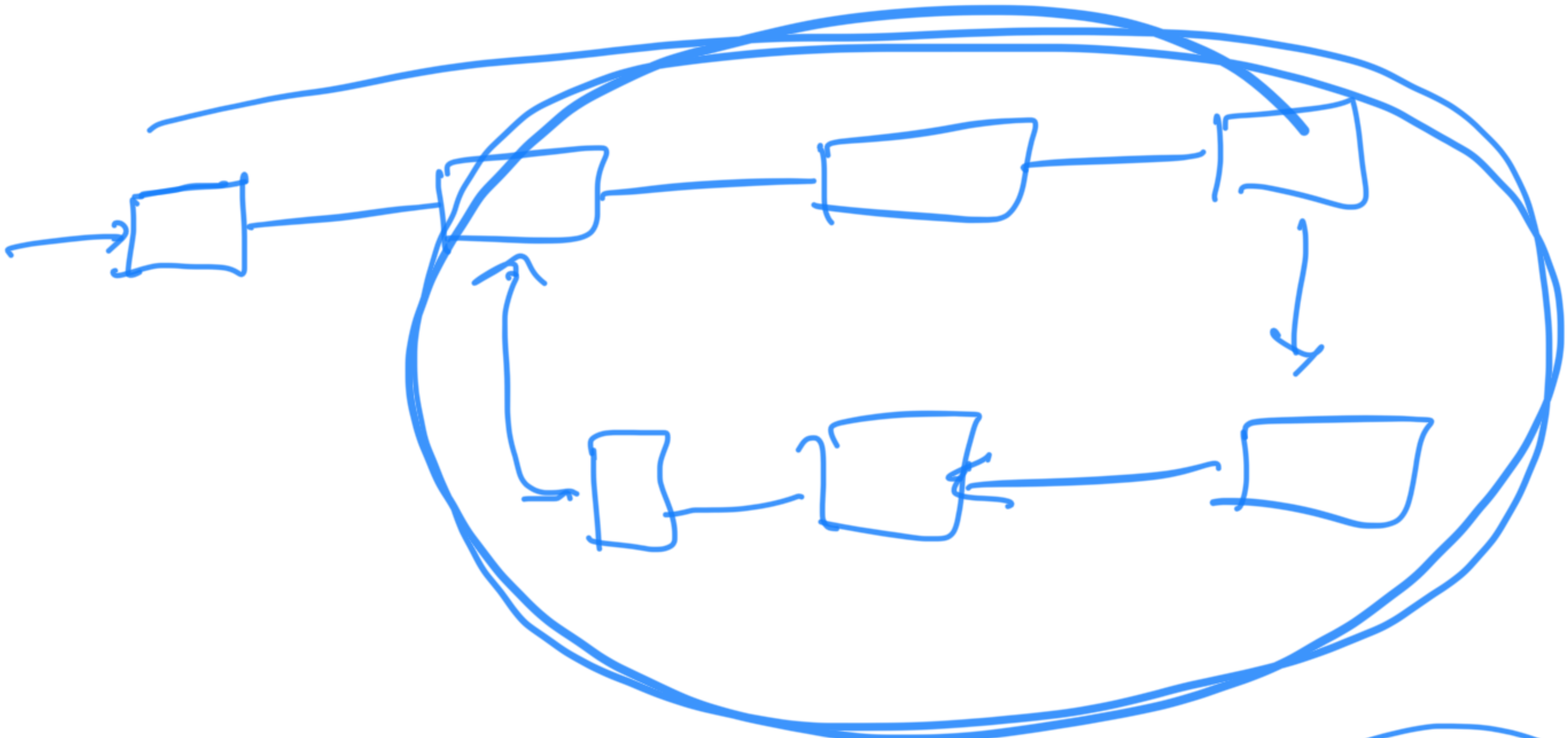
return prev;

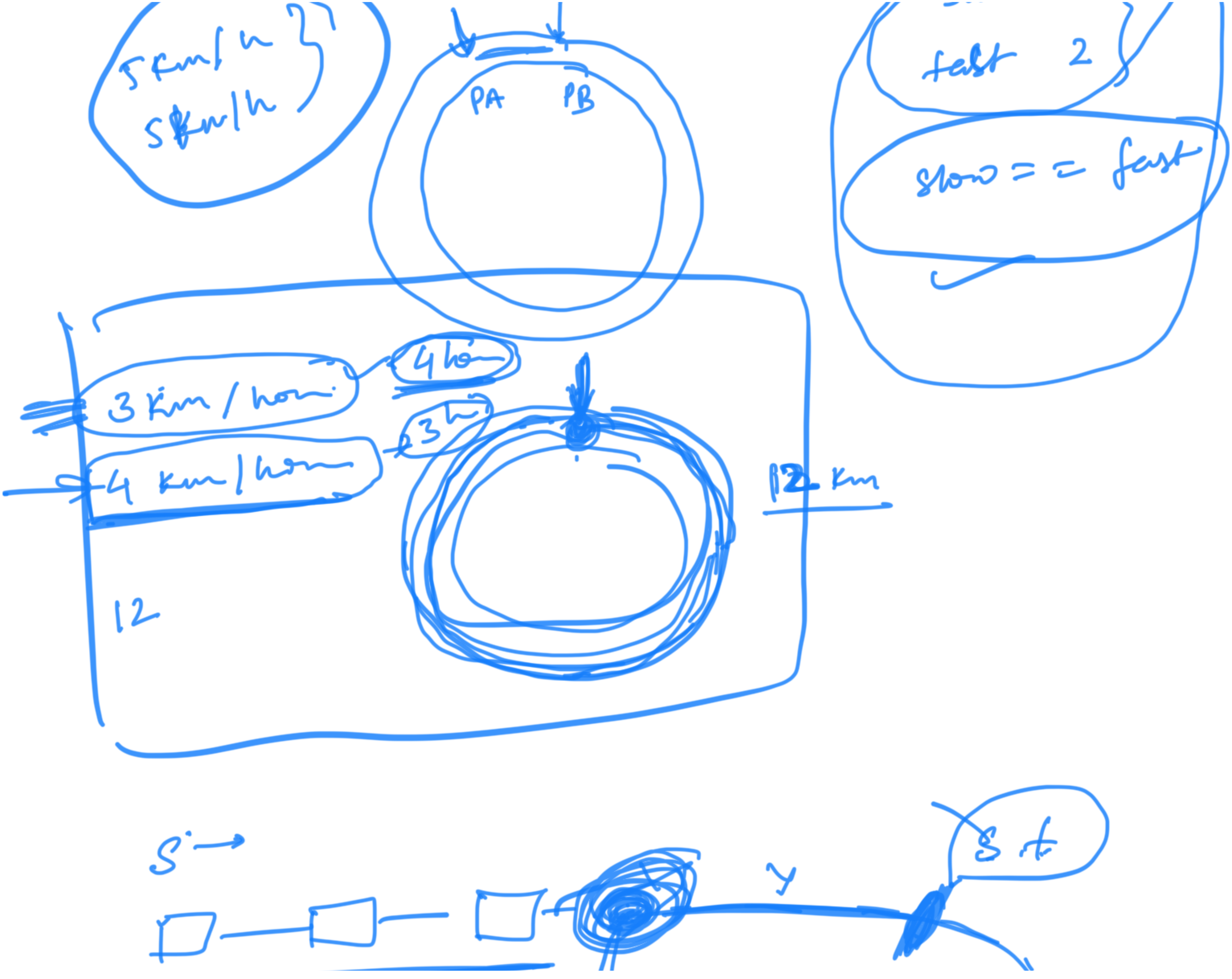
}

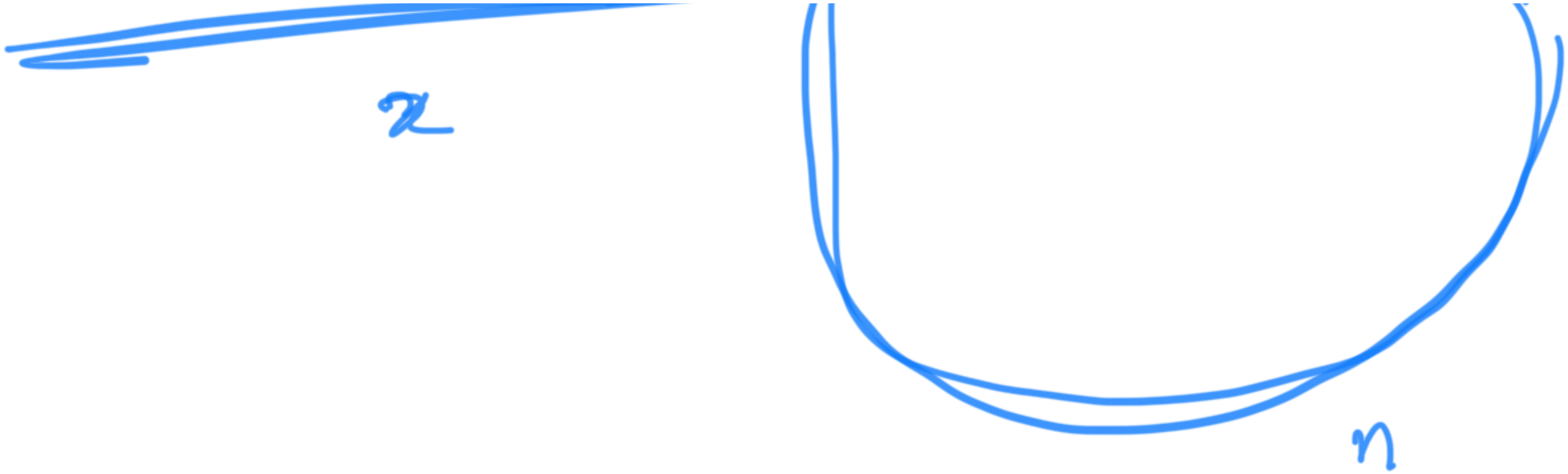
slow and fast pointer

floyd warshall algorithm

hare on tortoise




5km/h
5km/h
PA
PB
fast 2
slow == fast
3 km/hour
4 km/hour
4 hou
3 h
12 km
12
S
y
8 pt



~~find the~~

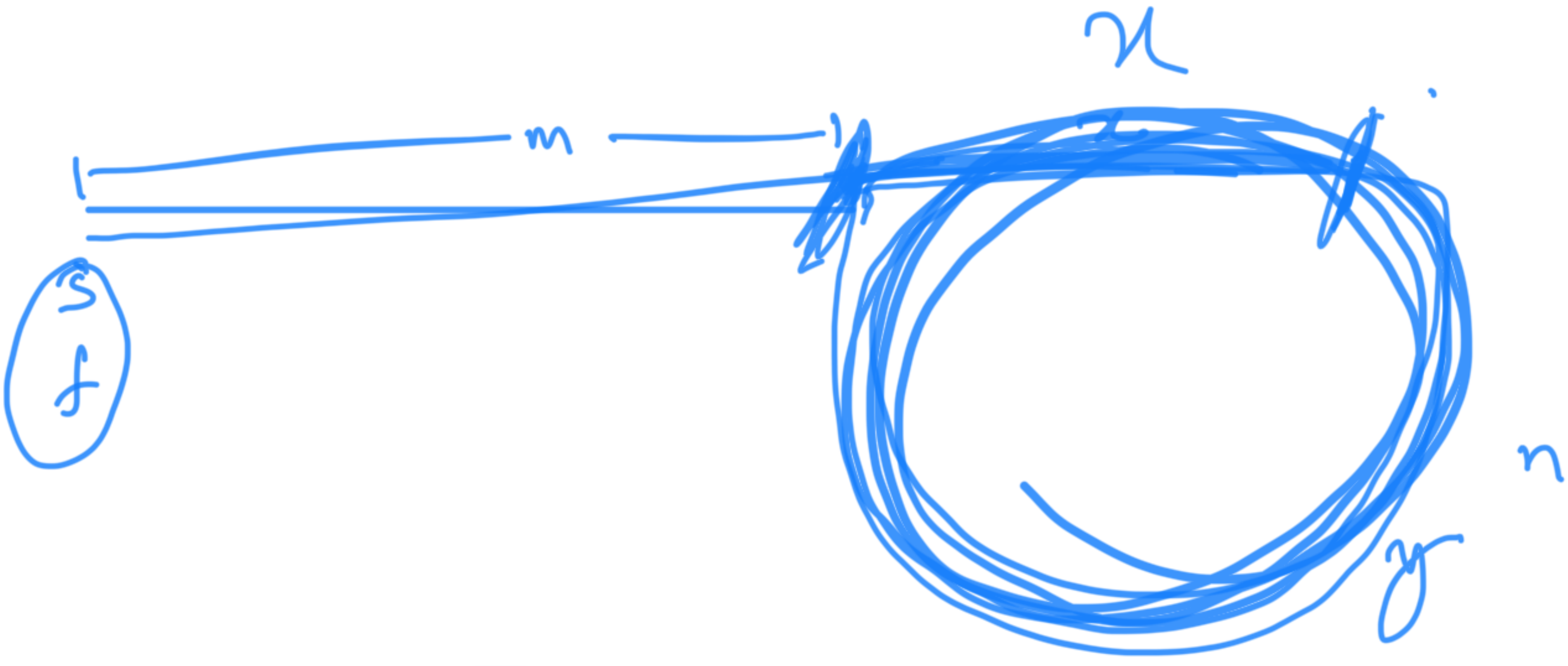
check the loop length.

if loop exists

~~reset~~ slow to head
and update both slow and
fast pointer 1 by 1.

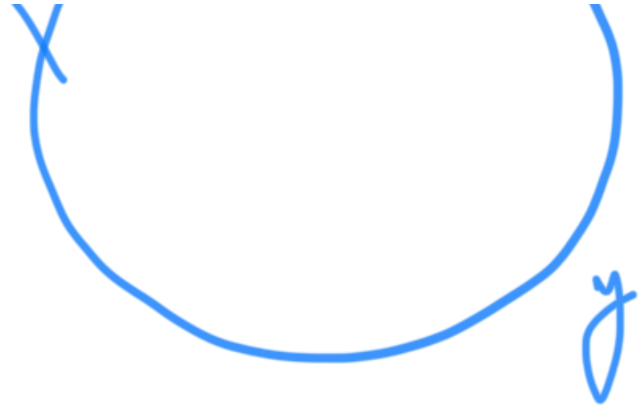
the place when they meet
again . is the starting

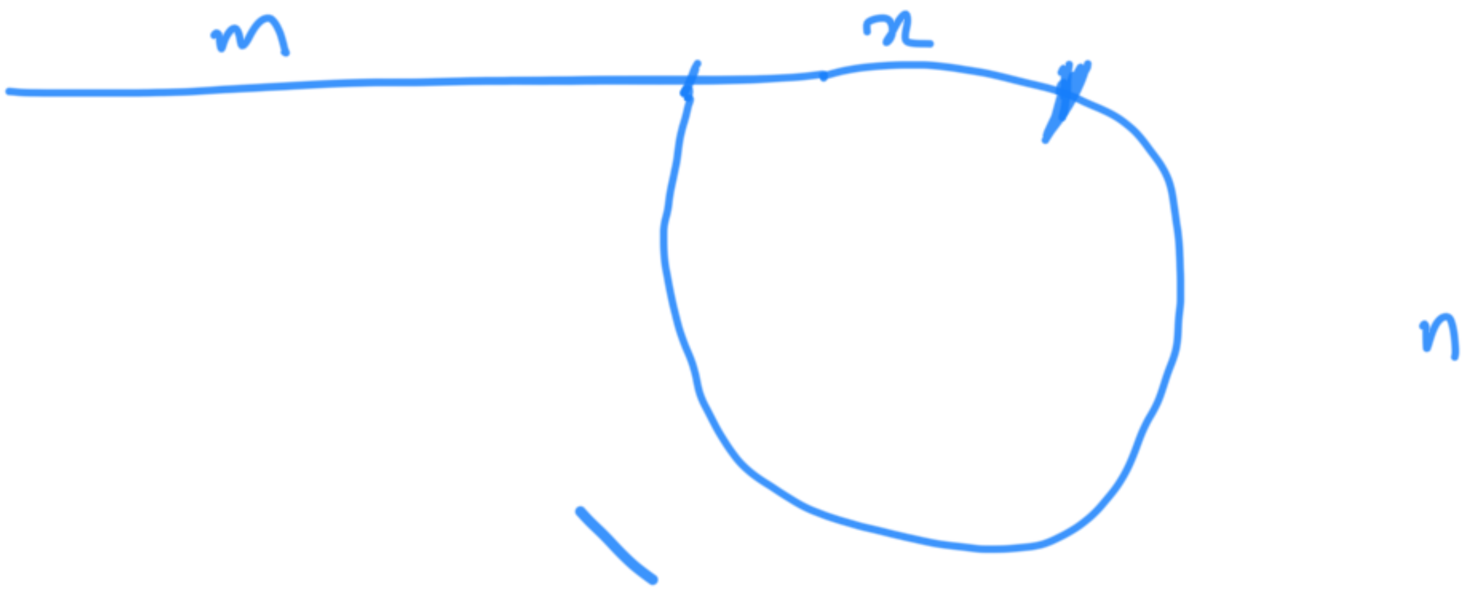
point of loop.



$$s = m + kn \rightarrow m \overset{y}{+} k(x+y)$$

$$f = 2(m + kn) \qquad 2(m + k(x+y))$$





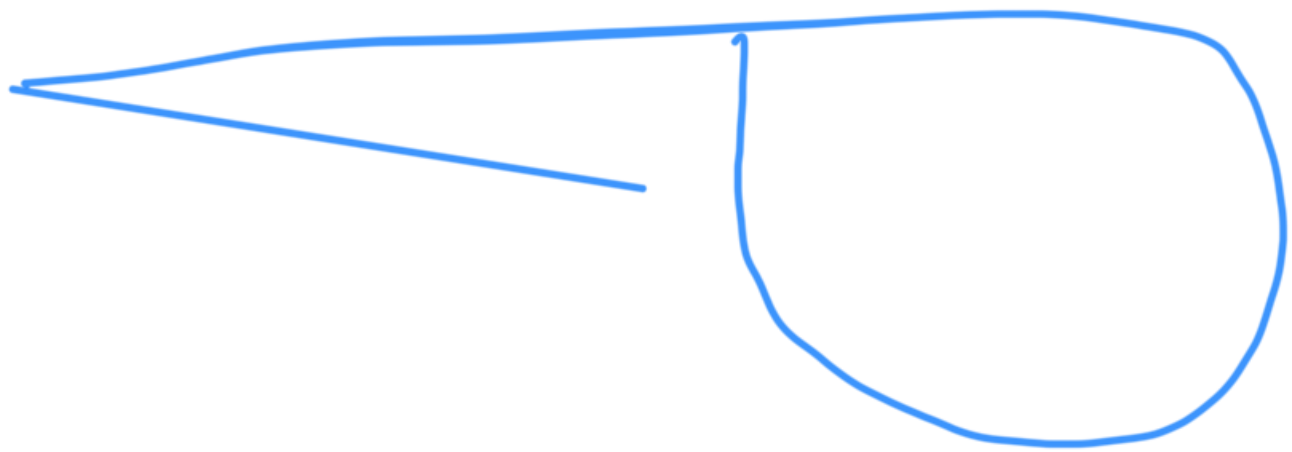
K_1 $\rightarrow$ ~~slow loop~~

K_2 $\rightarrow$ fast

How many time slow looped the circle

$m + k_1 * n$ $m + k_2 * n$

$\cancel{m + x}$



$s = m$

$f = n - x$

$$\boxed{m + k_1 \times n + x}$$ slow pointer

$$\boxed{m + k_2 \times n + x}$$ fast pointer

$$m + k_2 \times n + x = 2(m + k_1 \times n + x)$$

$$m + x - 2m - 2x = 2k_1 n - k_2 n$$

$$-(m + x) = n(2k_1 - k_2)$$

$$\boxed{m + x = n(k_2 - 2k_1)}$$ 1

$$\boxed{m = n - x}$$

$m + x = n$

$m + x =$ multiple of n

$n - x$

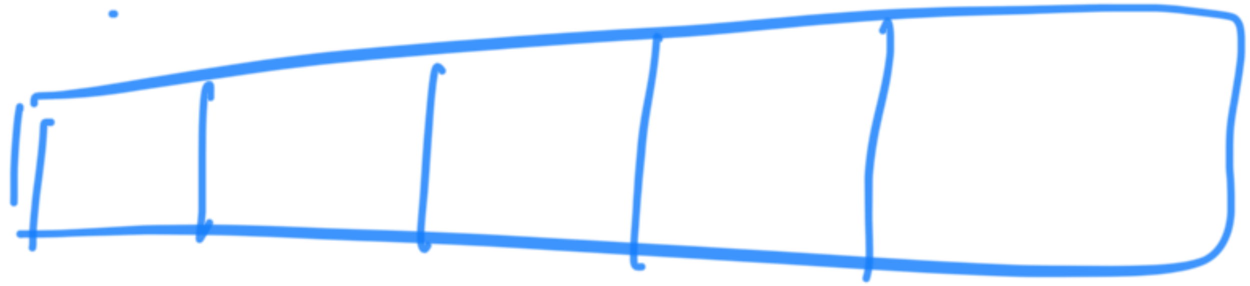
$$\boxed{K_3 \times n + n - x}$$

$$\boxed{m = K_3 \times n + n - x}$$

$$\boxed{m + x = n(K_3 + 1)}$$

3, 4 $\Rightarrow$ 12

N



1, N-1

N [1, N-1]

[1, 4]

5

[1, 5-1]

[1, 4]

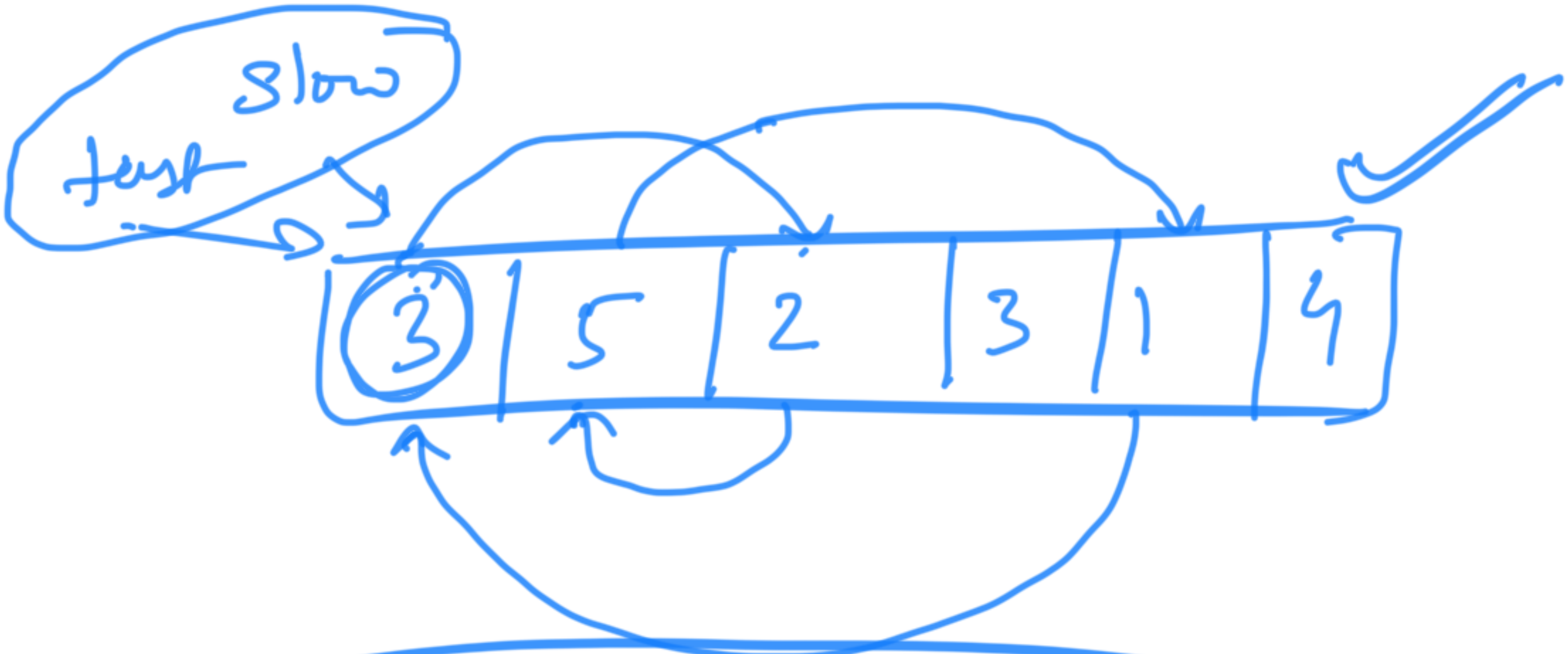
[1, 4]

Given an array of size N allowed to contain element in range $[1, N-1]$ such array is bound to have repeatation.

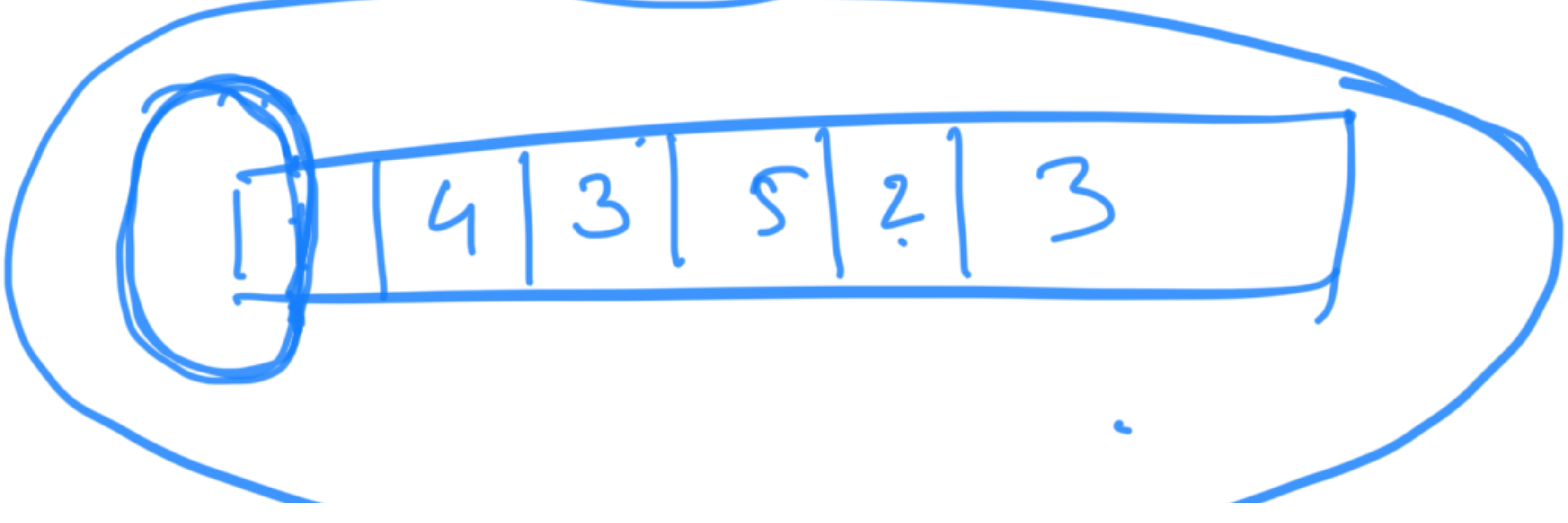
2	2	2	1	1

$[1, 4]$

4	3	1	2	4



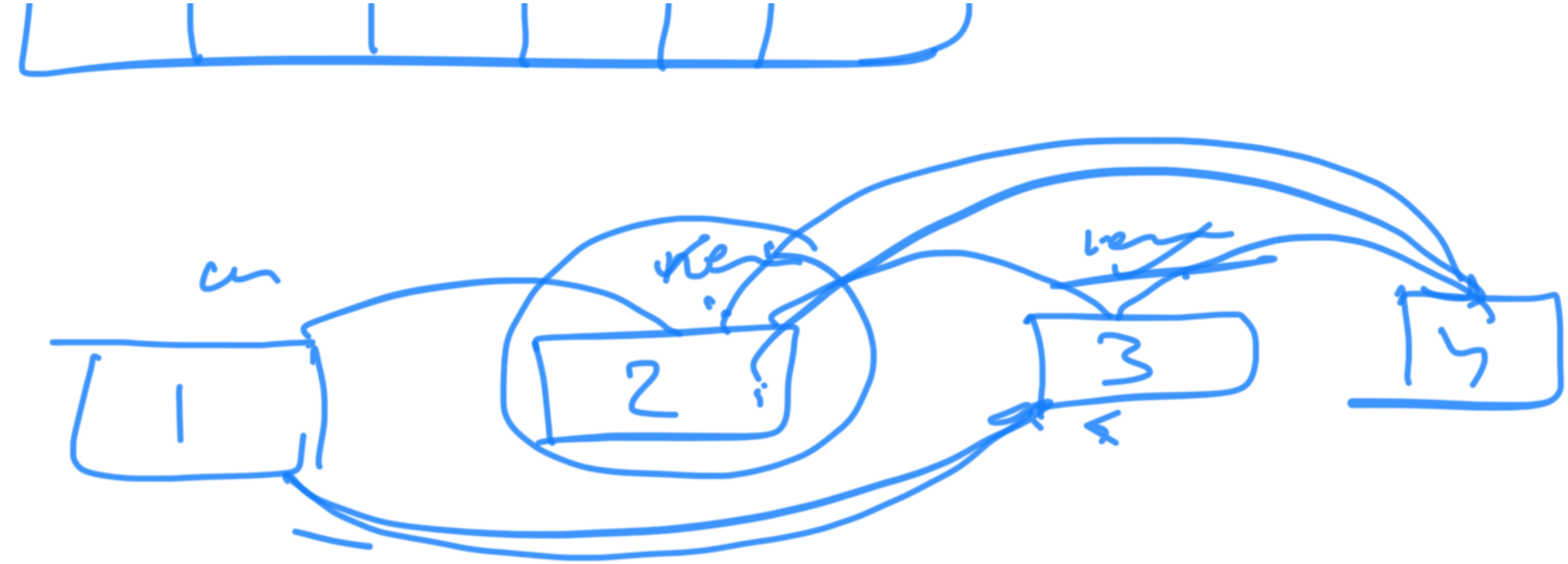
[1, 5]



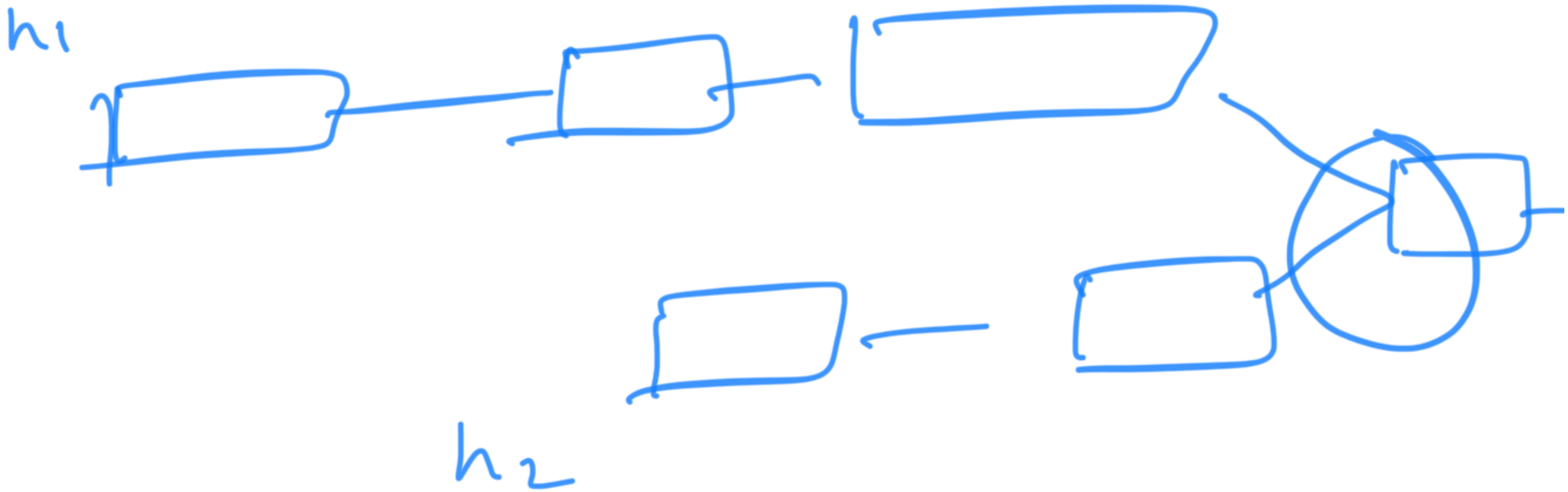
slow →
fast →

slow = head
fast

slow ++
fast ++



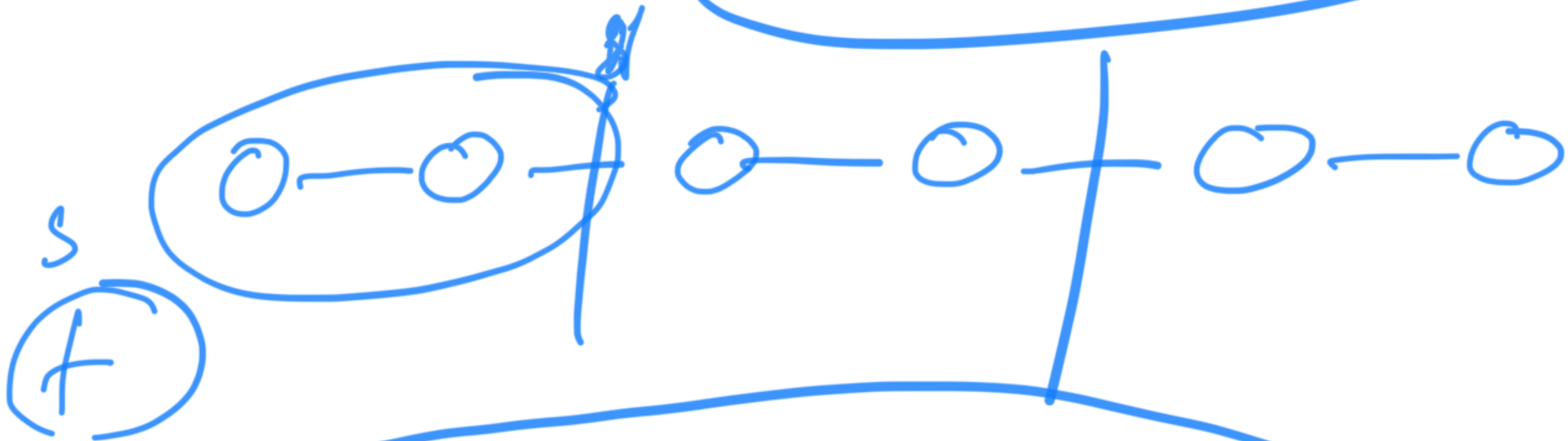
1
2
3
4

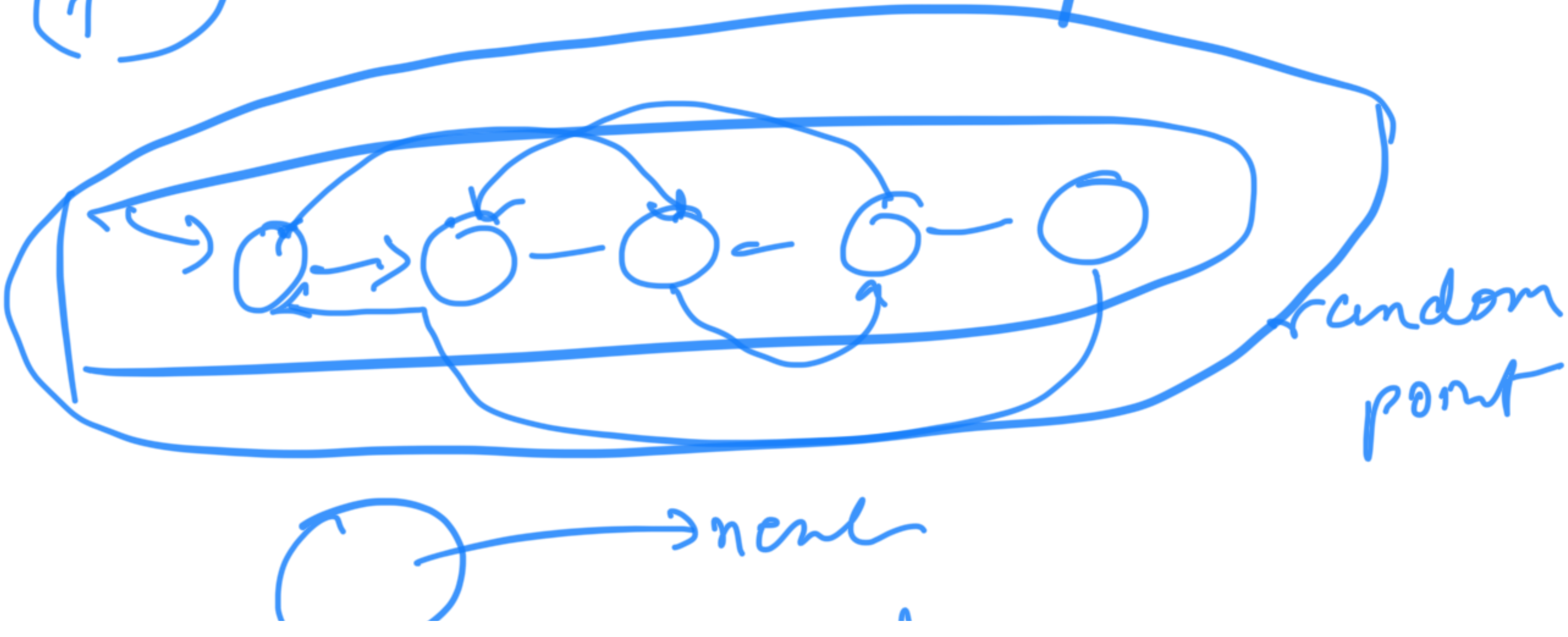


h1
h2

S


s
f
slow fast


s
f


random
point
next

→ approach

LRU Cache

Linked HashMap ✓

Linked List and hash Map