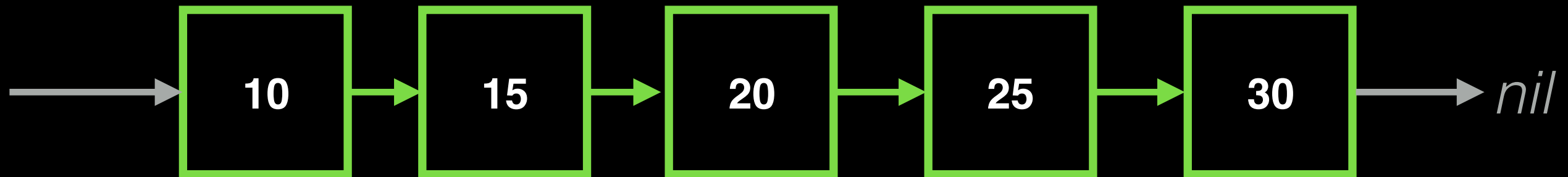


# Fine-Grained & Lock-Free Skip List

Alec Bargas, Julian Domingo, Jacob Ingalls, John Starich

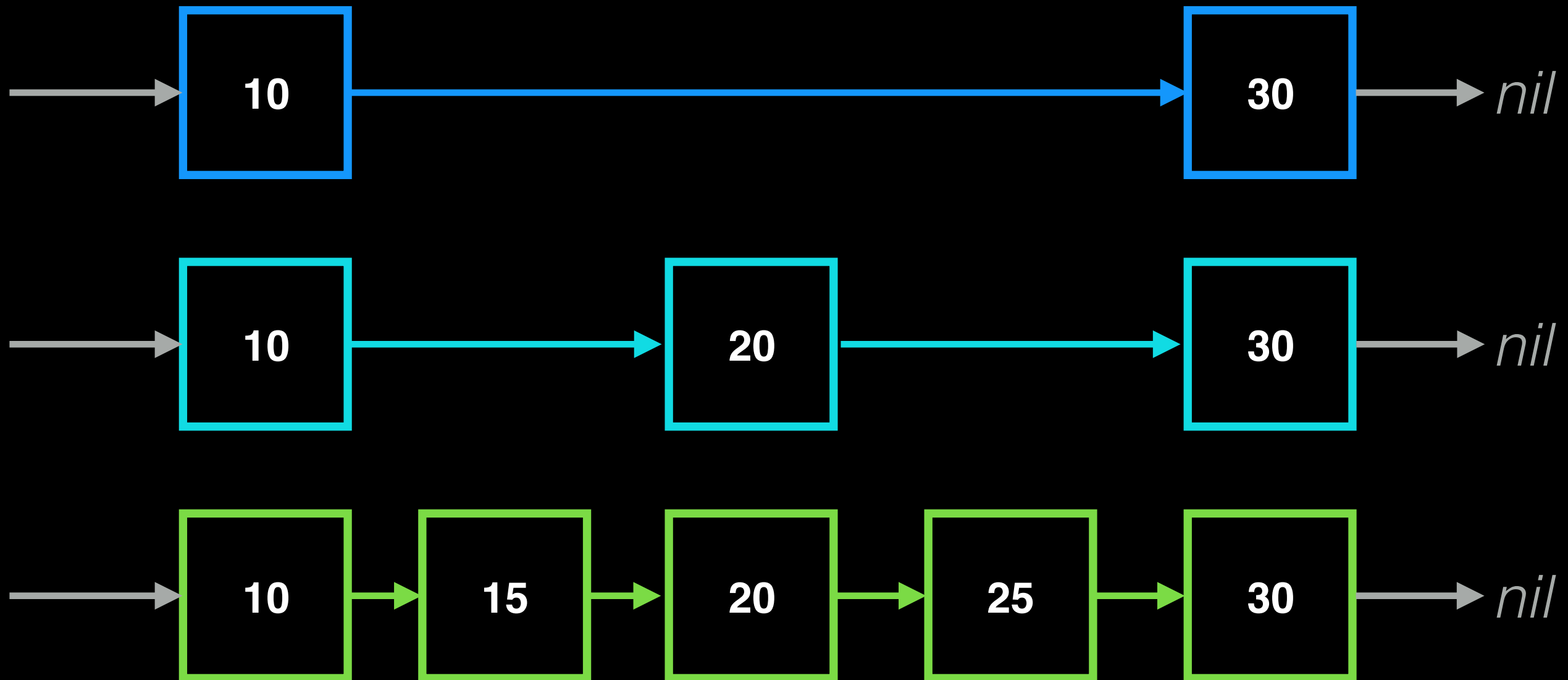
# A what?

It is like a **Sorted, Tiered** Linked List



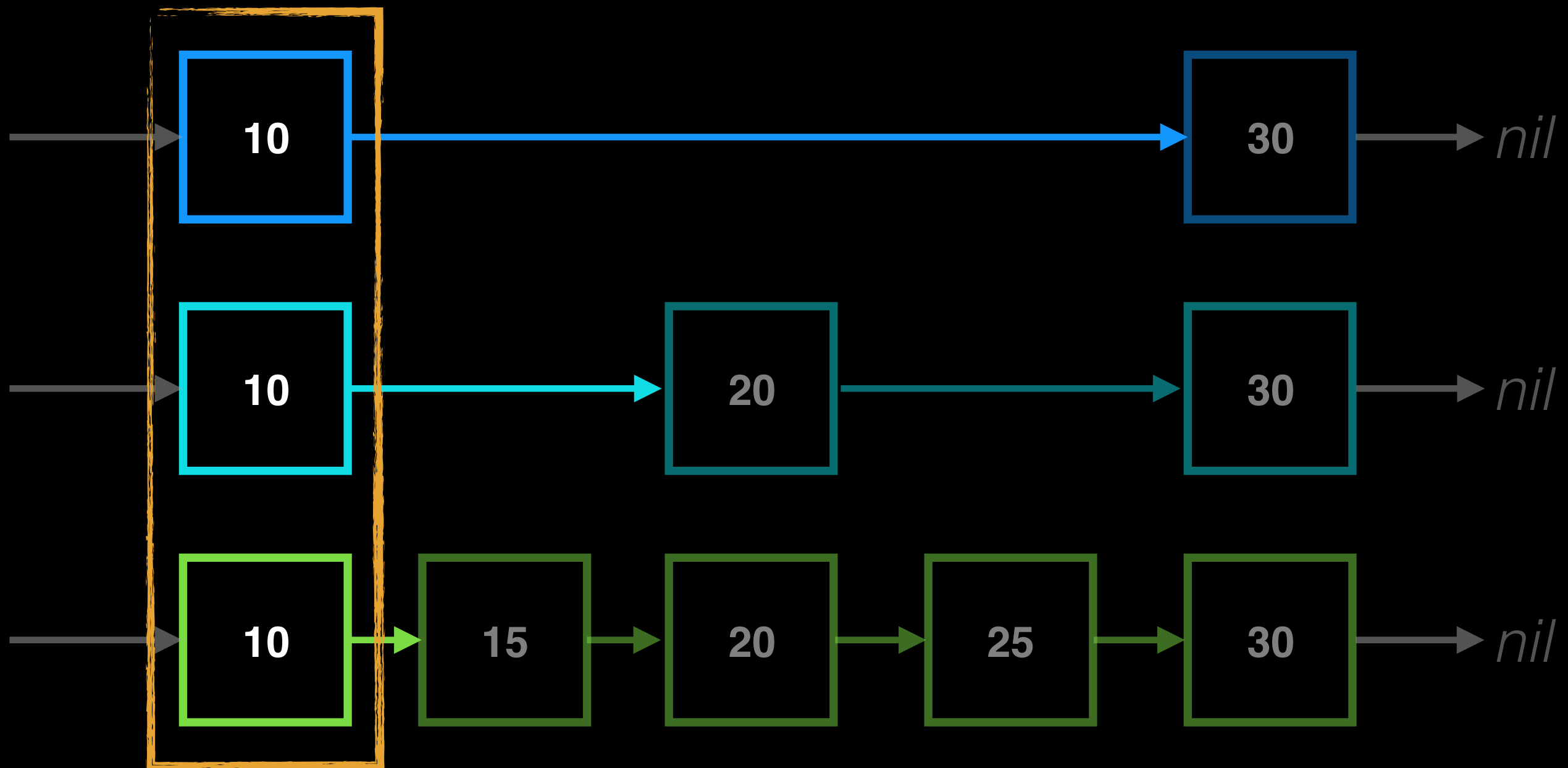
# A what?

It is like a **Sorted, Tiered** Linked List



# A what?

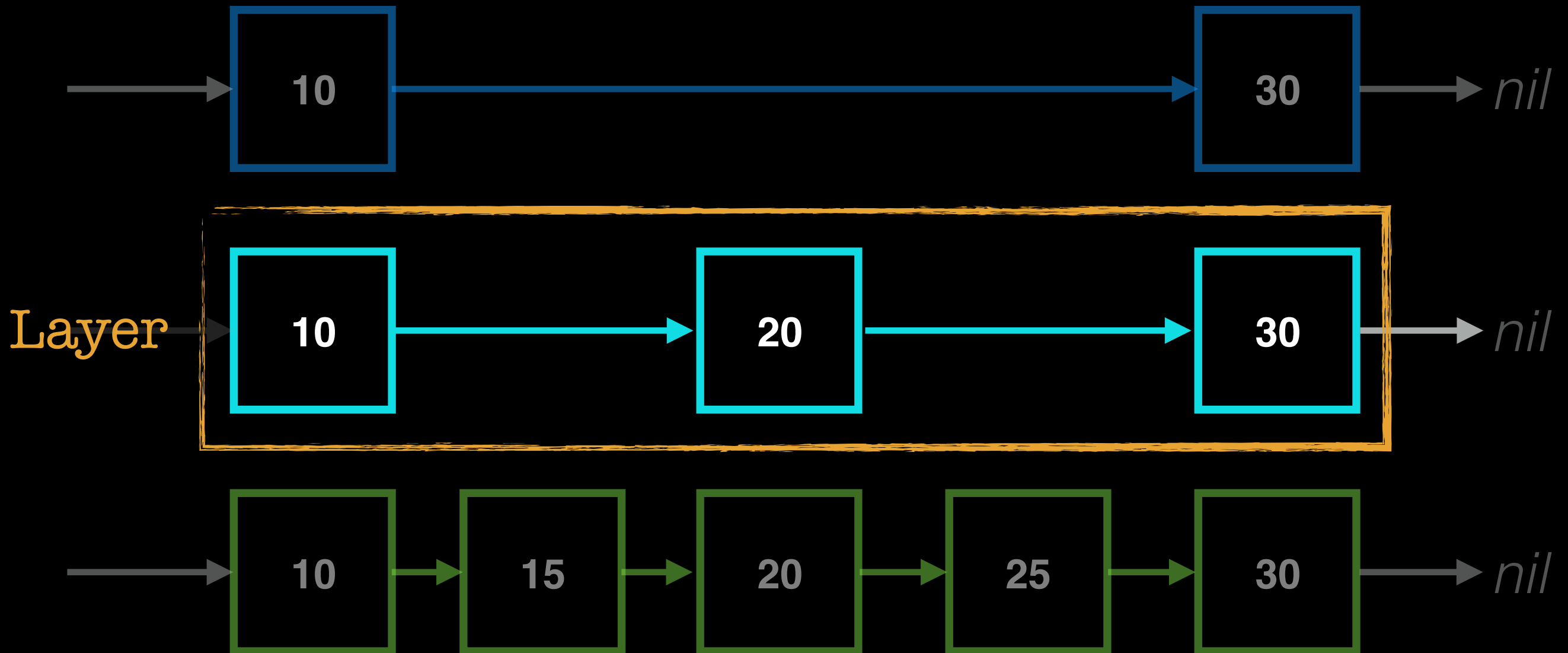
It is like a **Sorted, Tiered** Linked List



Node

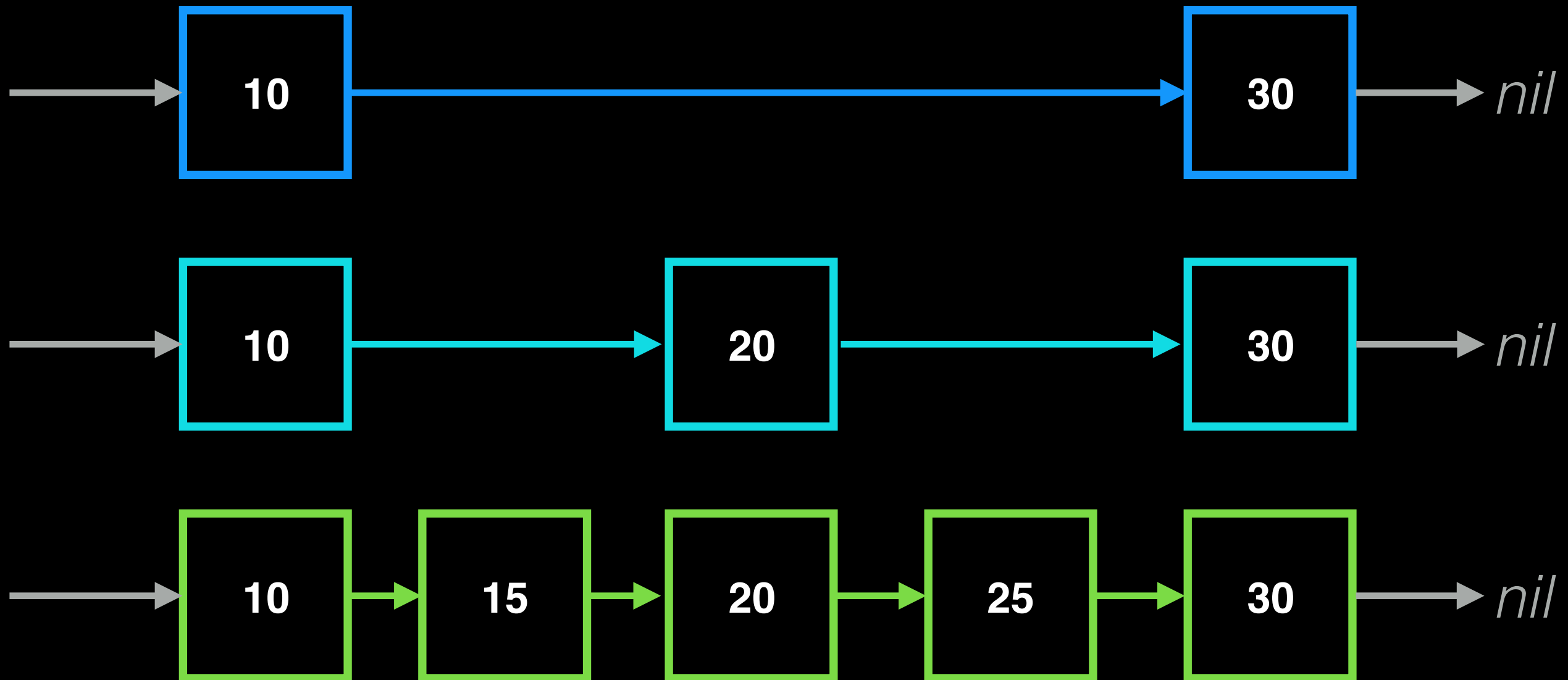
# A what?

It is like a **Sorted, Tiered** Linked List



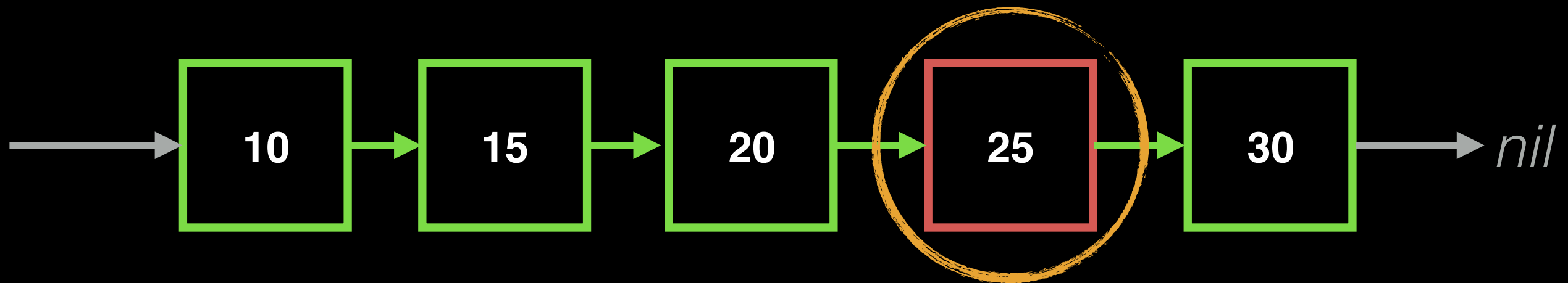
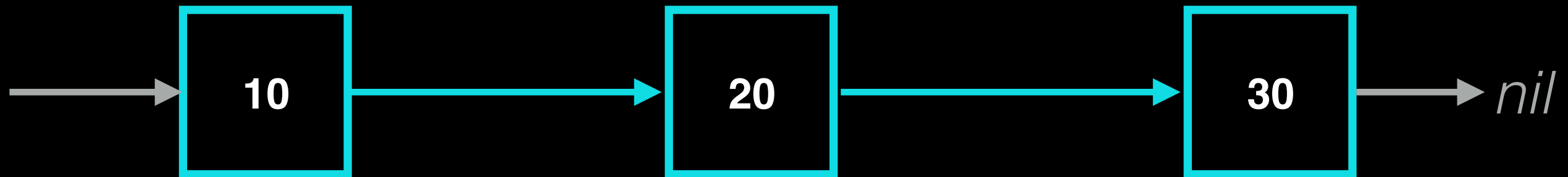
# A what?

It is like a **Sorted, Tiered** Linked List



# Search

$\Theta(\text{Log}(n))$

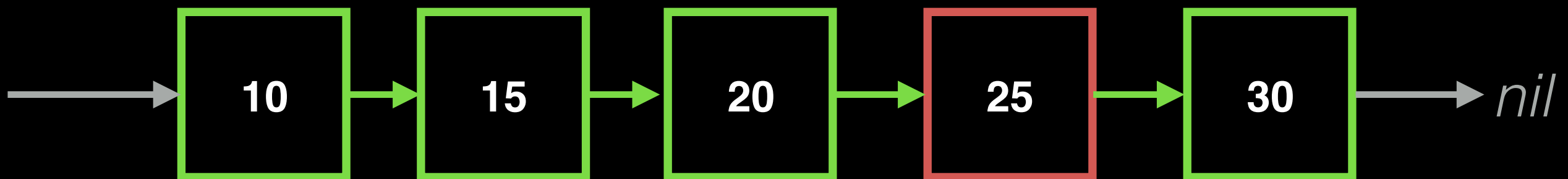
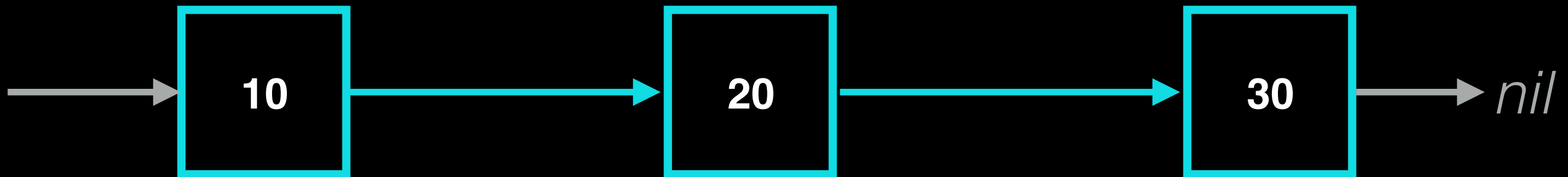
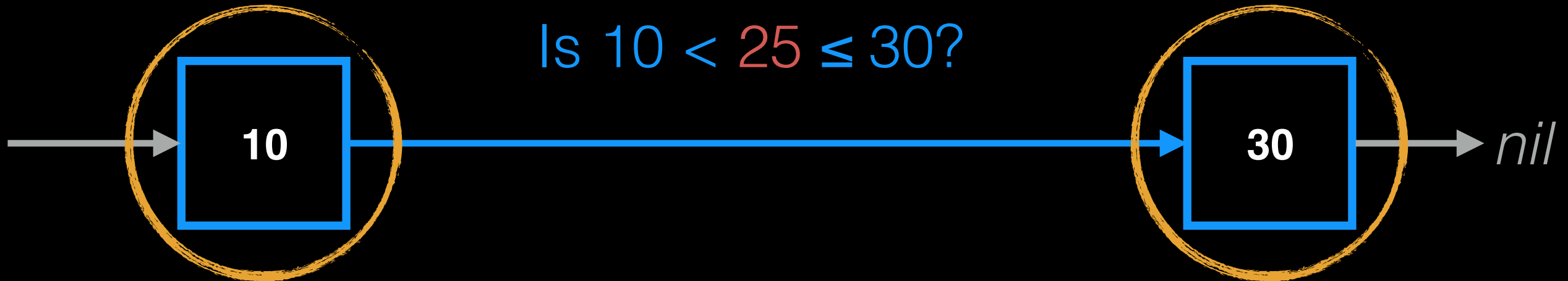


Target

# Search

$\Theta(\log(n))$

Is  $10 < 25 \leq 30$ ?



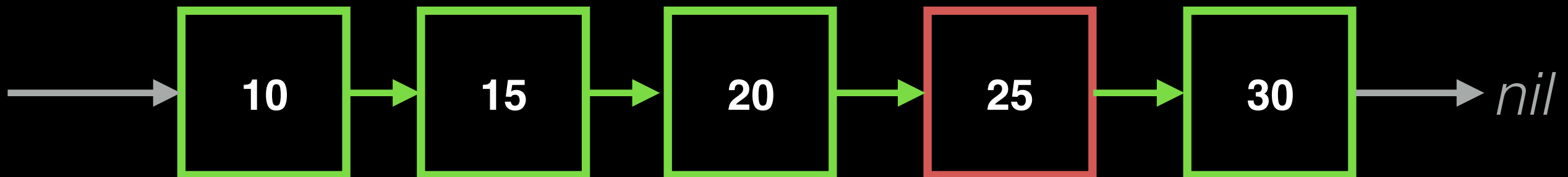
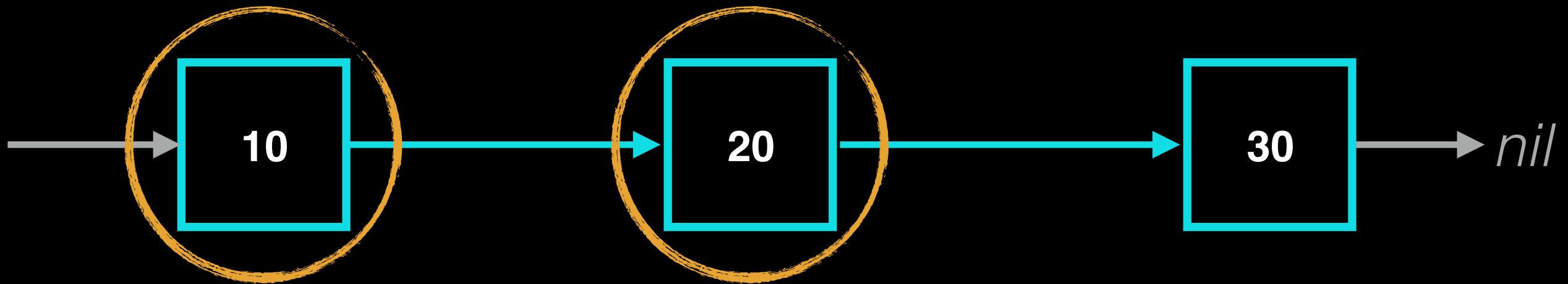
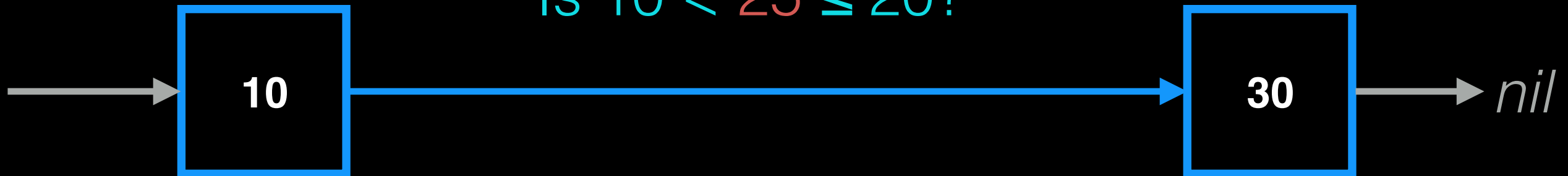
Target



# Search

$\Theta(\log(n))$

Is  $10 < 25 \leq 20$ ?

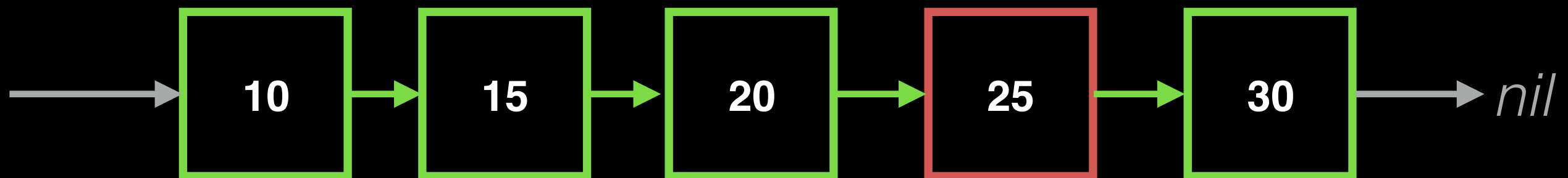
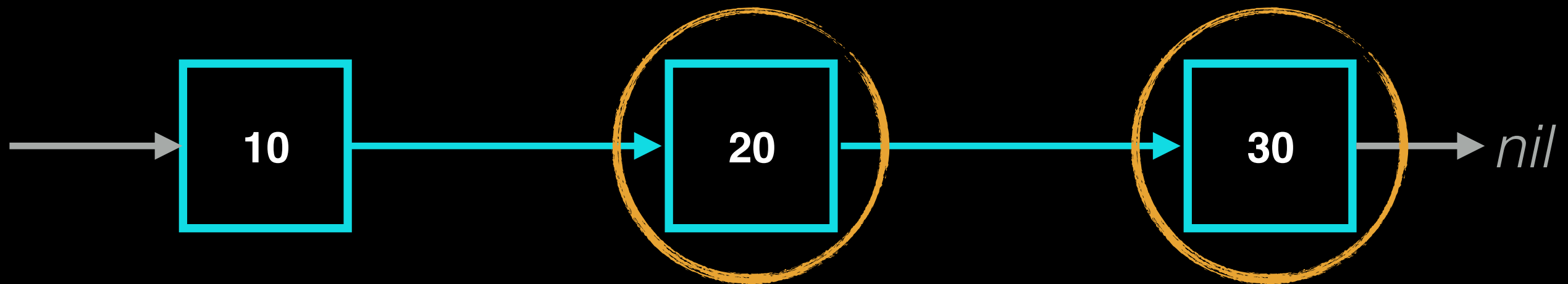
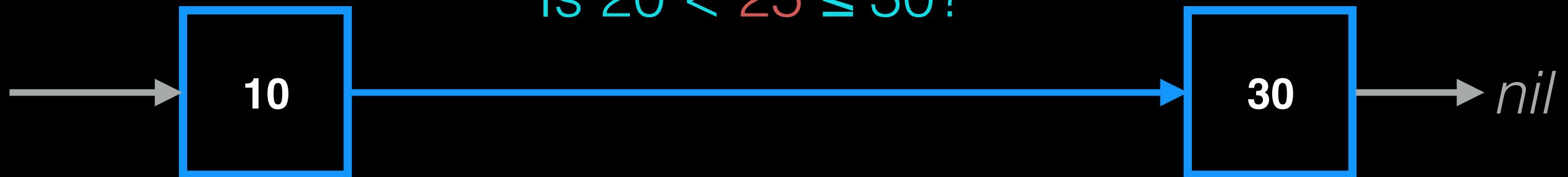


Target

# Search

$\Theta(\log(n))$

Is  $20 < 25 \leq 30$ ?

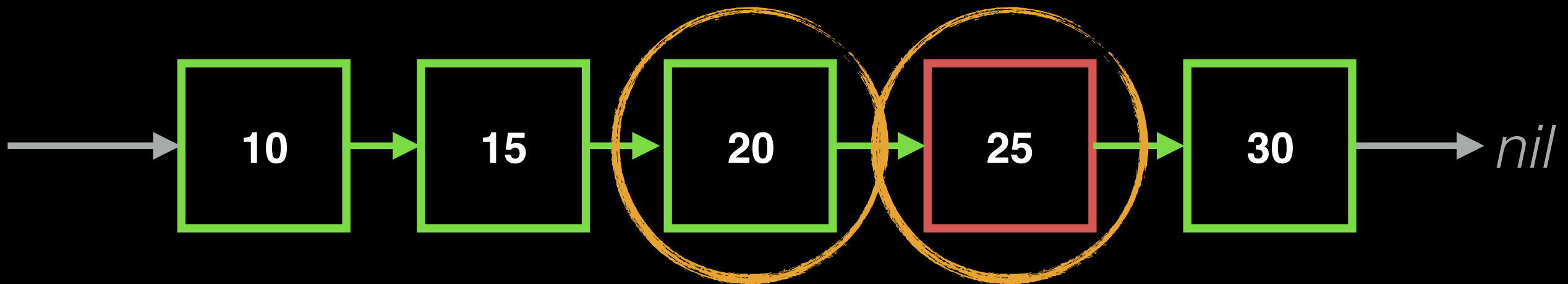
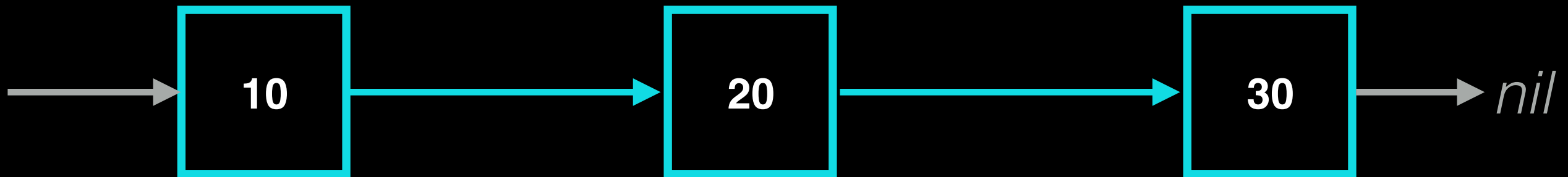
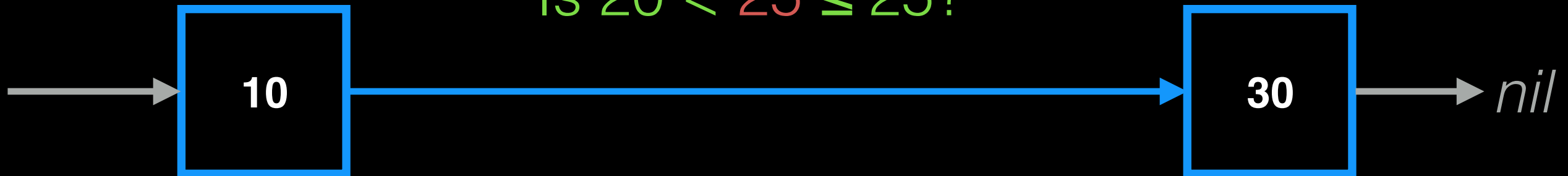


Target

# Search

$\Theta(\log(n))$

Is  $20 < 25 \leq 25$ ?

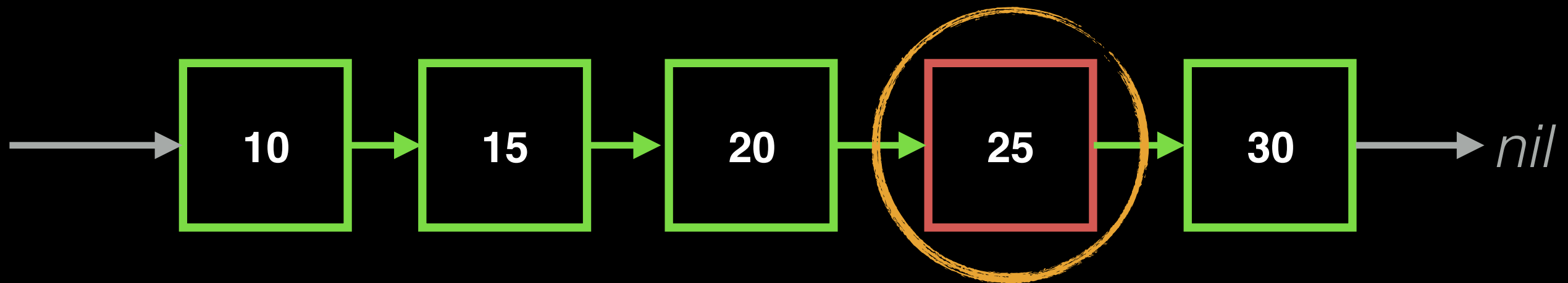
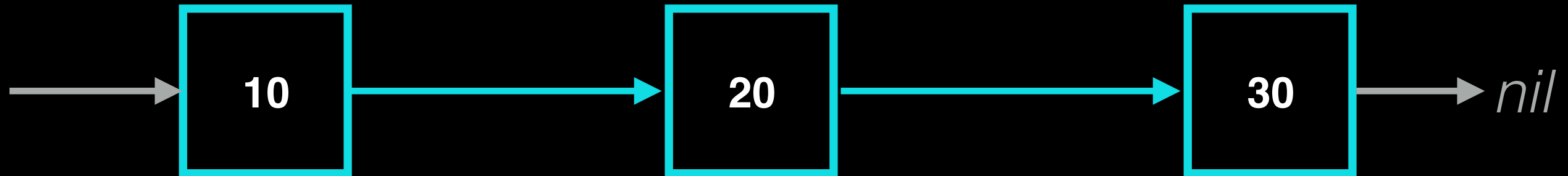
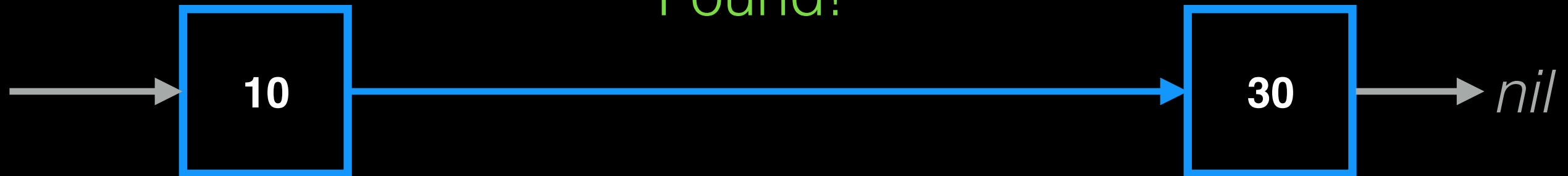


Target

# Search

$\Theta(\log(n))$

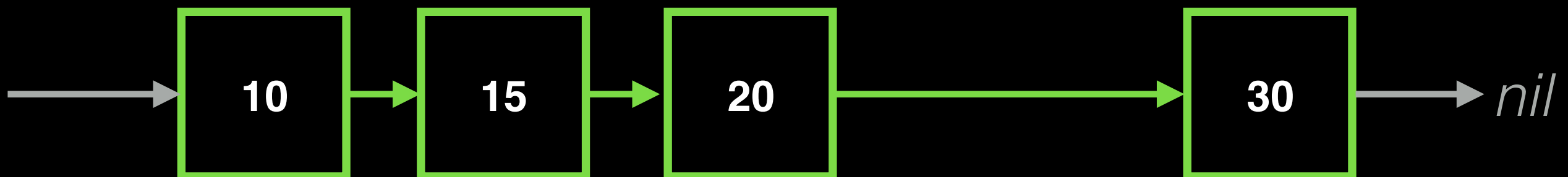
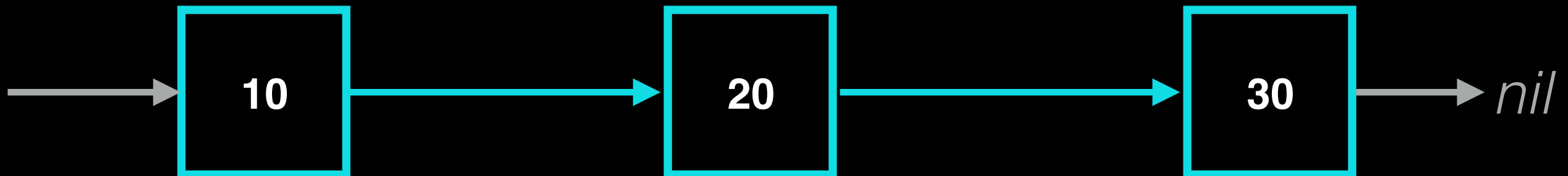
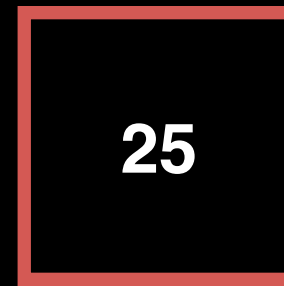
Found!



Target

# Insert

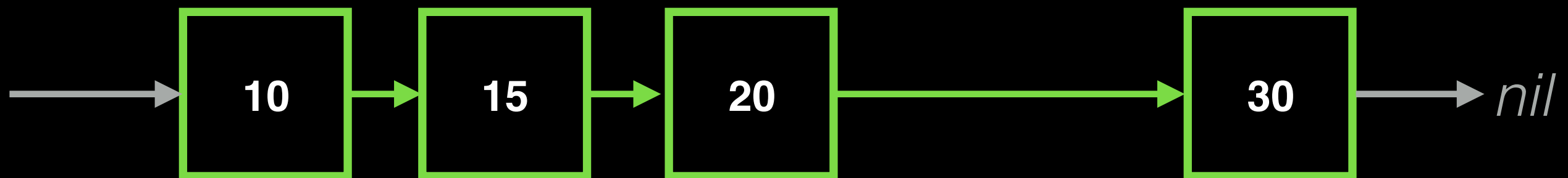
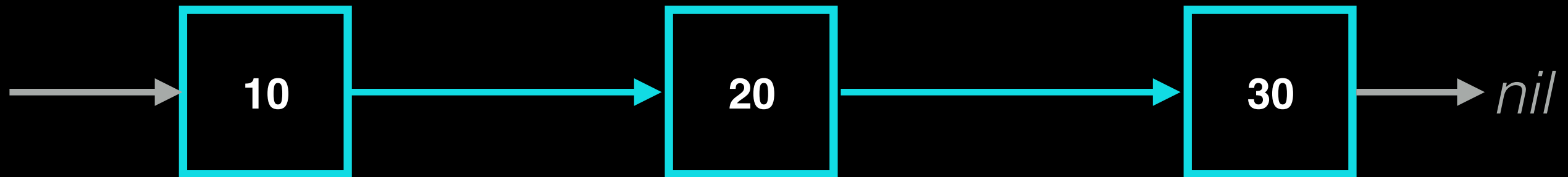
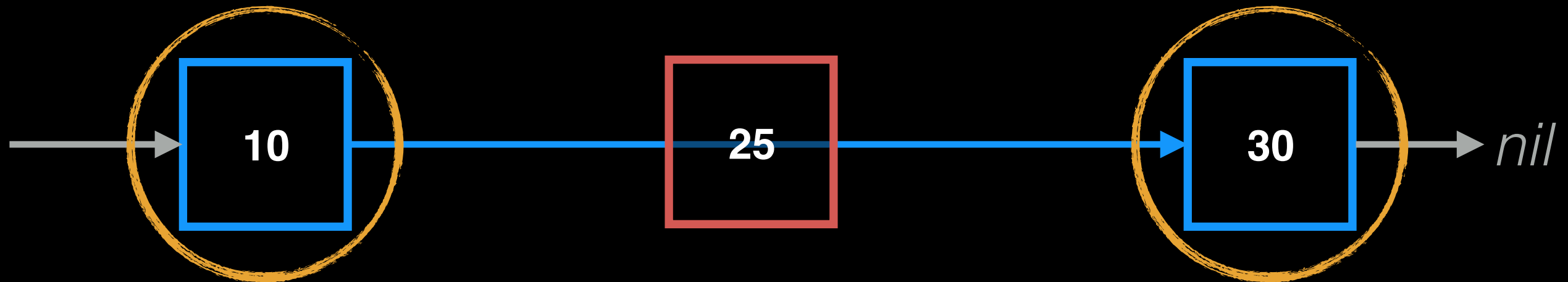
$\Theta(\text{Log}(n))$



Target

# Insert

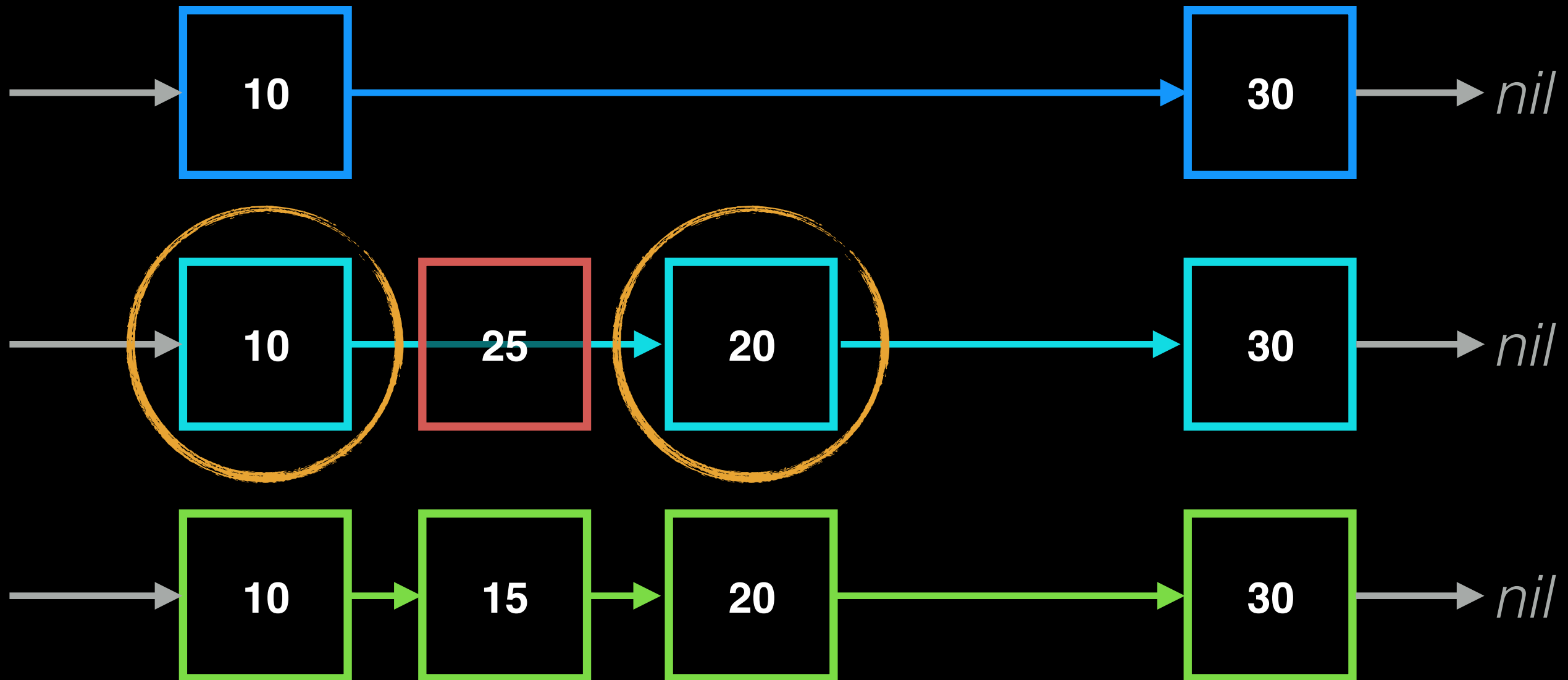
$\Theta(\log(n))$



Target

# Insert

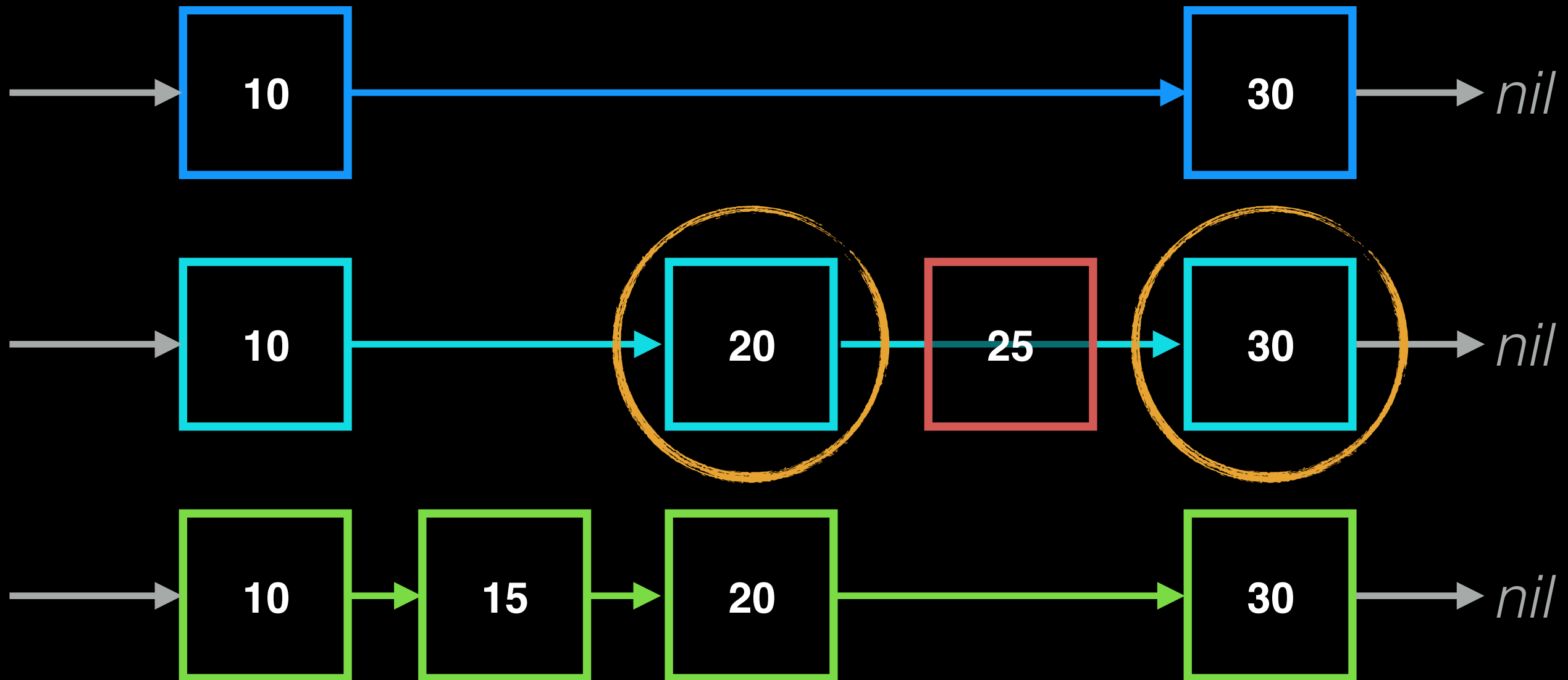
$\Theta(\text{Log}(n))$



Target

# Insert

$\Theta(\text{Log}(n))$

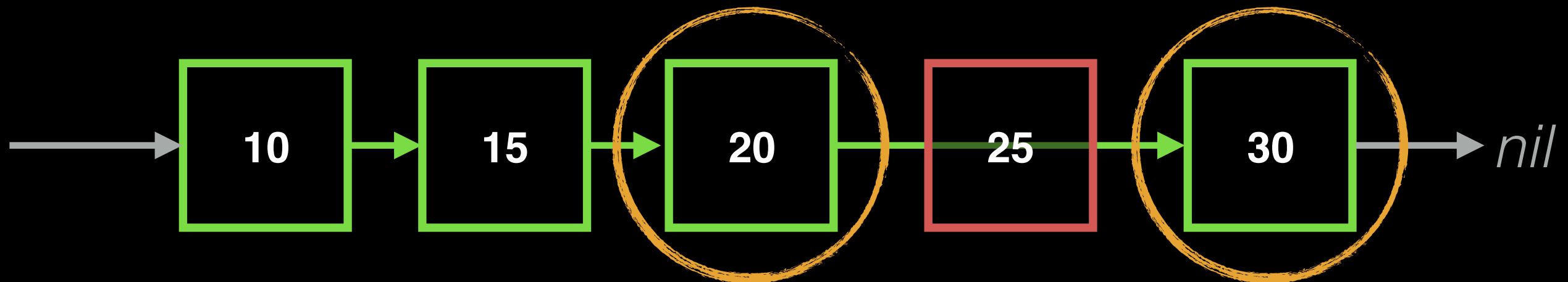
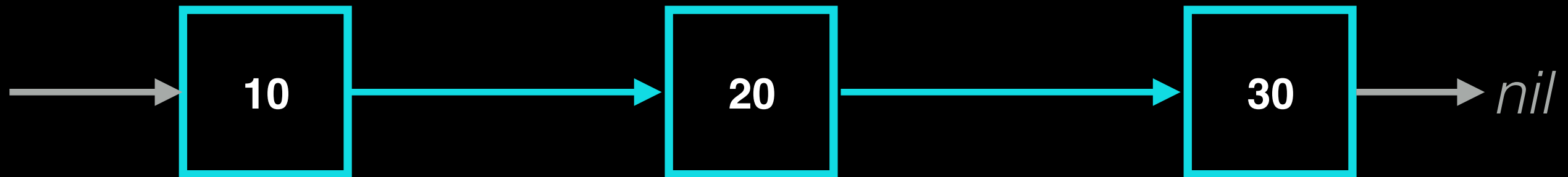
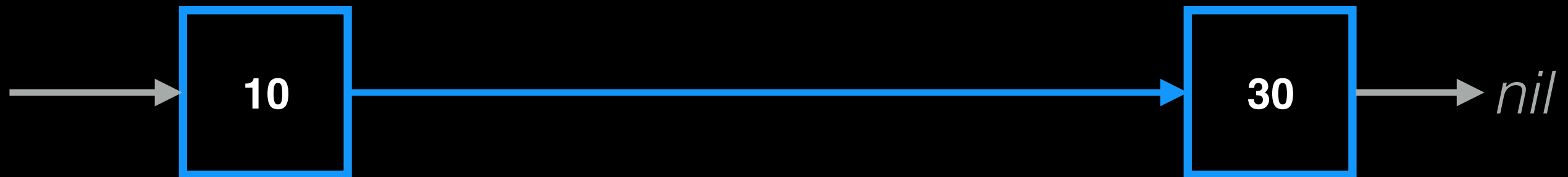


Target



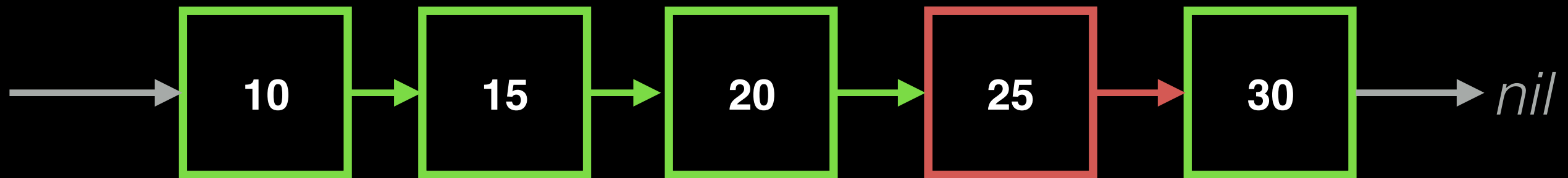
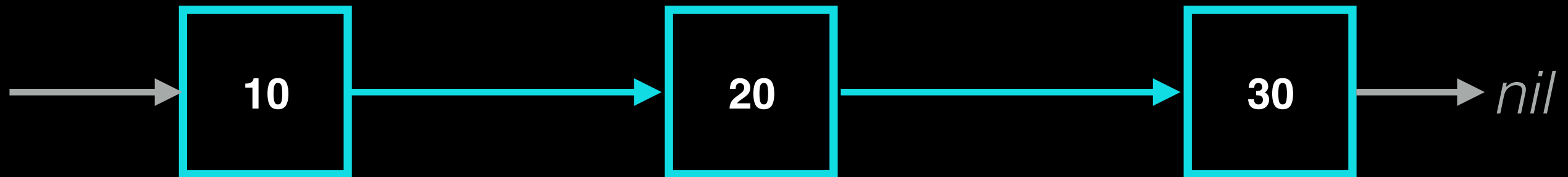
# Insert

$\Theta(\log(n))$



# Insert

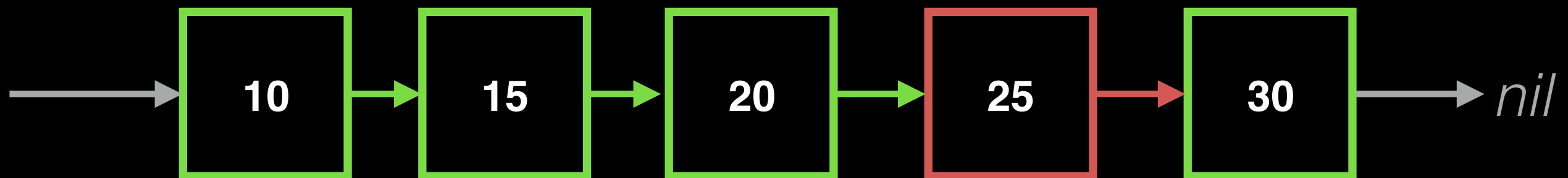
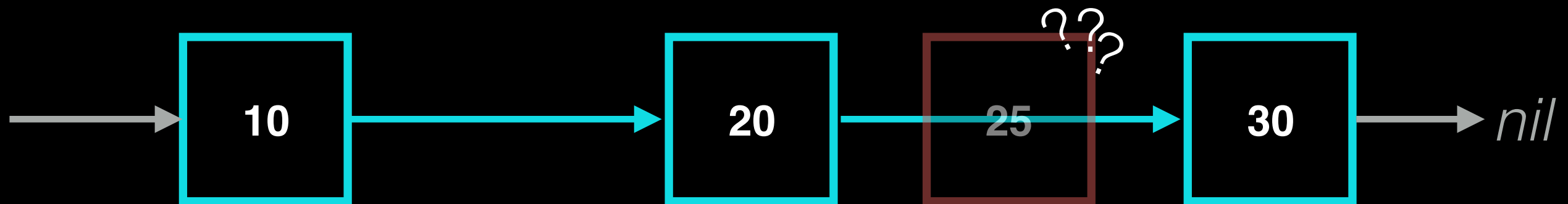
$\Theta(\log(n))$



Target

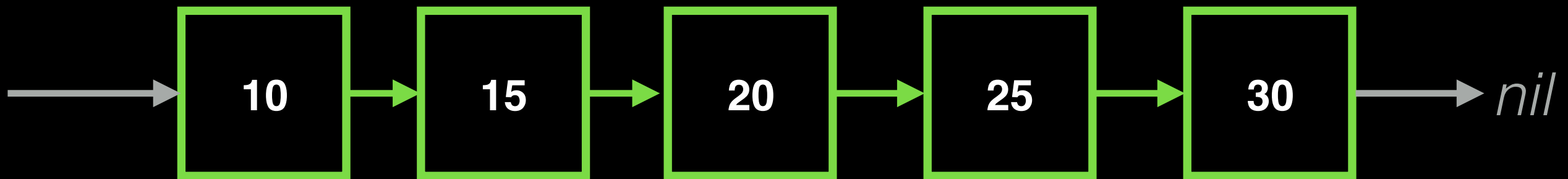
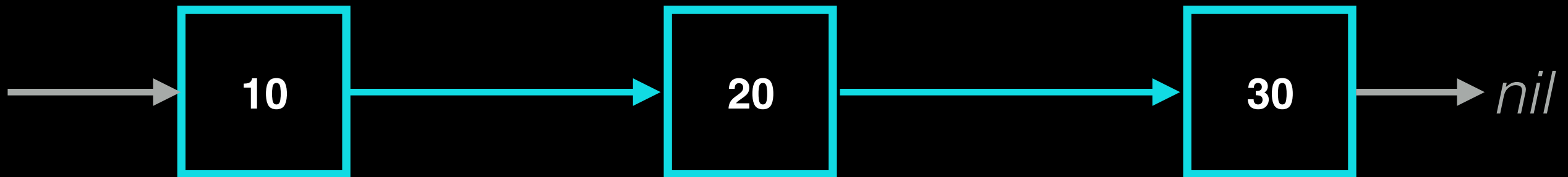
# Insert

$\Theta(\text{Log}(n))$



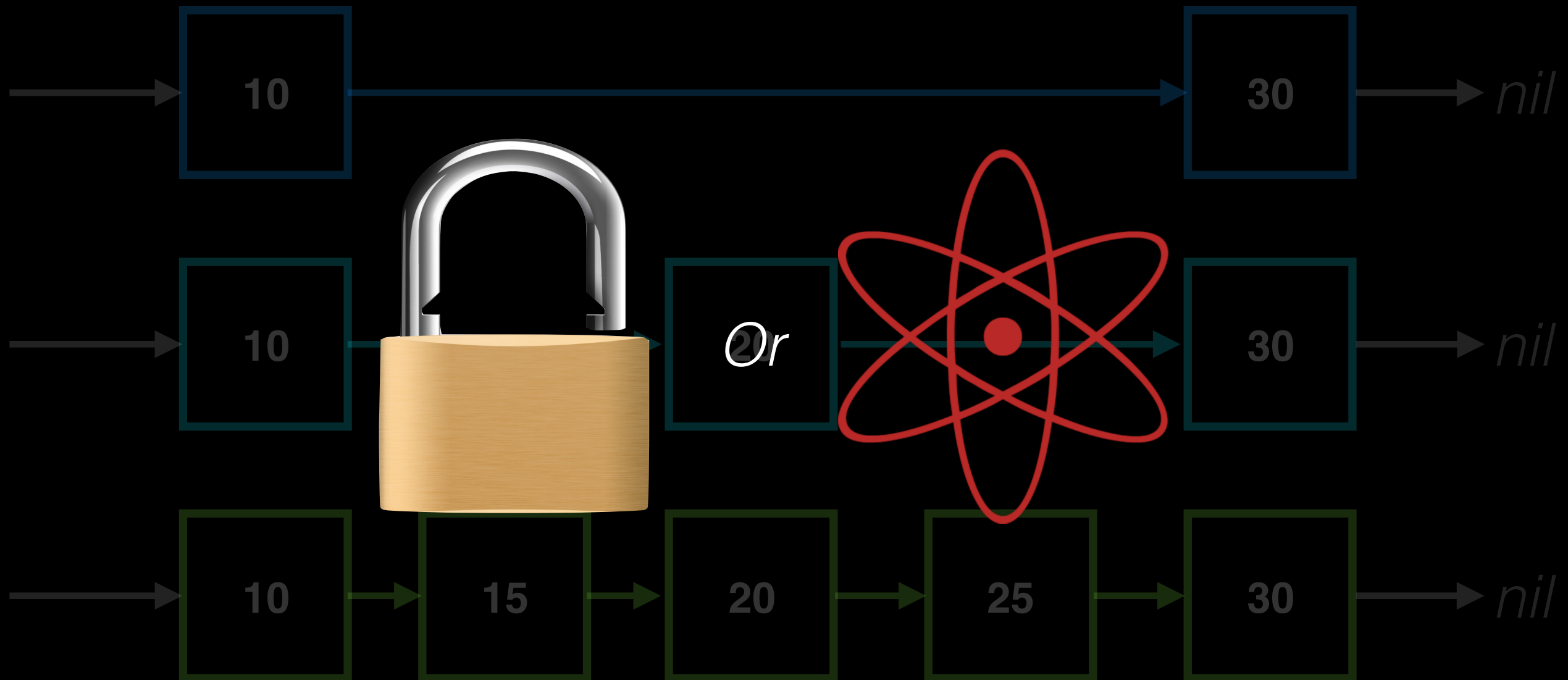
# Insert

$\Theta(\log(n))$



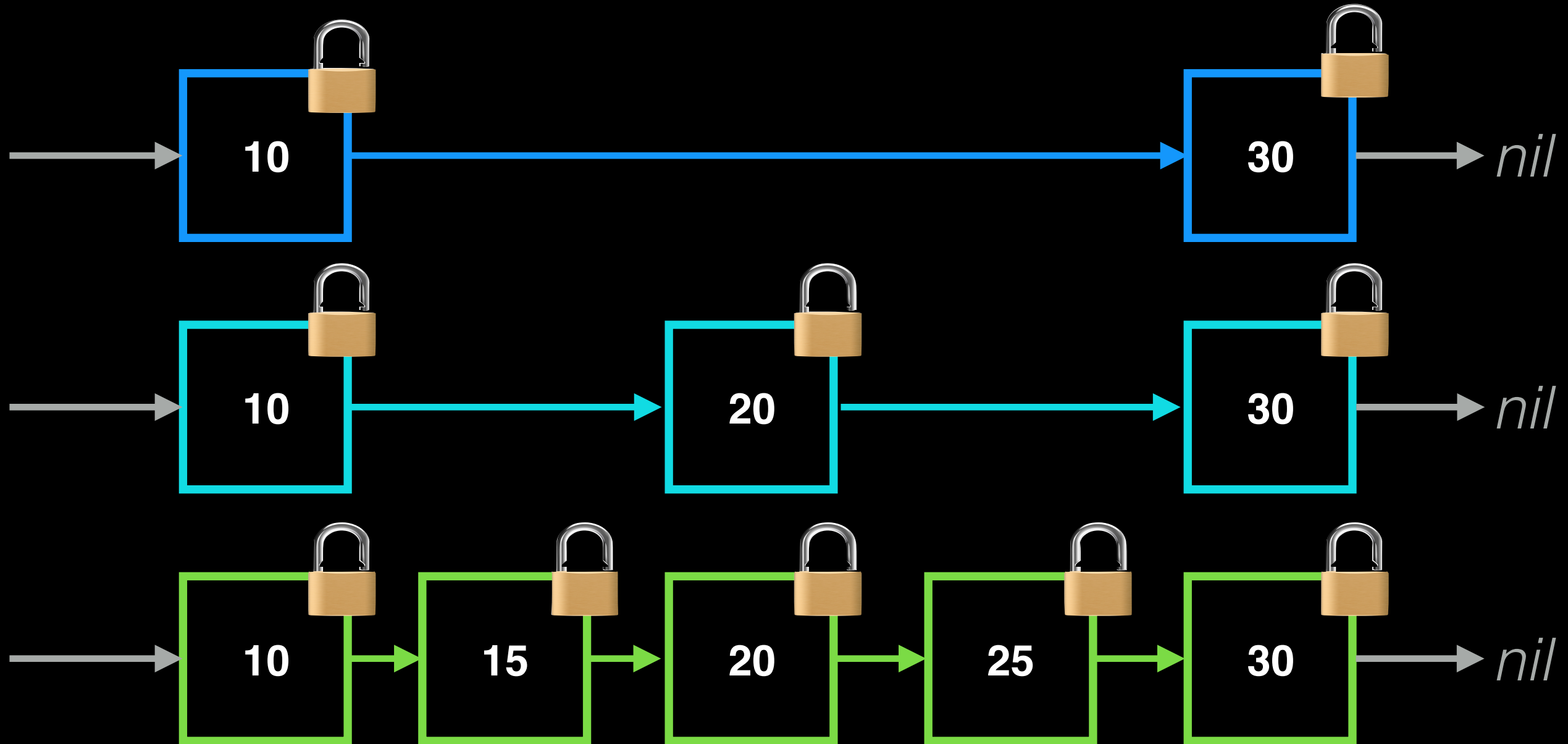
# Concurrency

Fine-Grained or Lock-Free



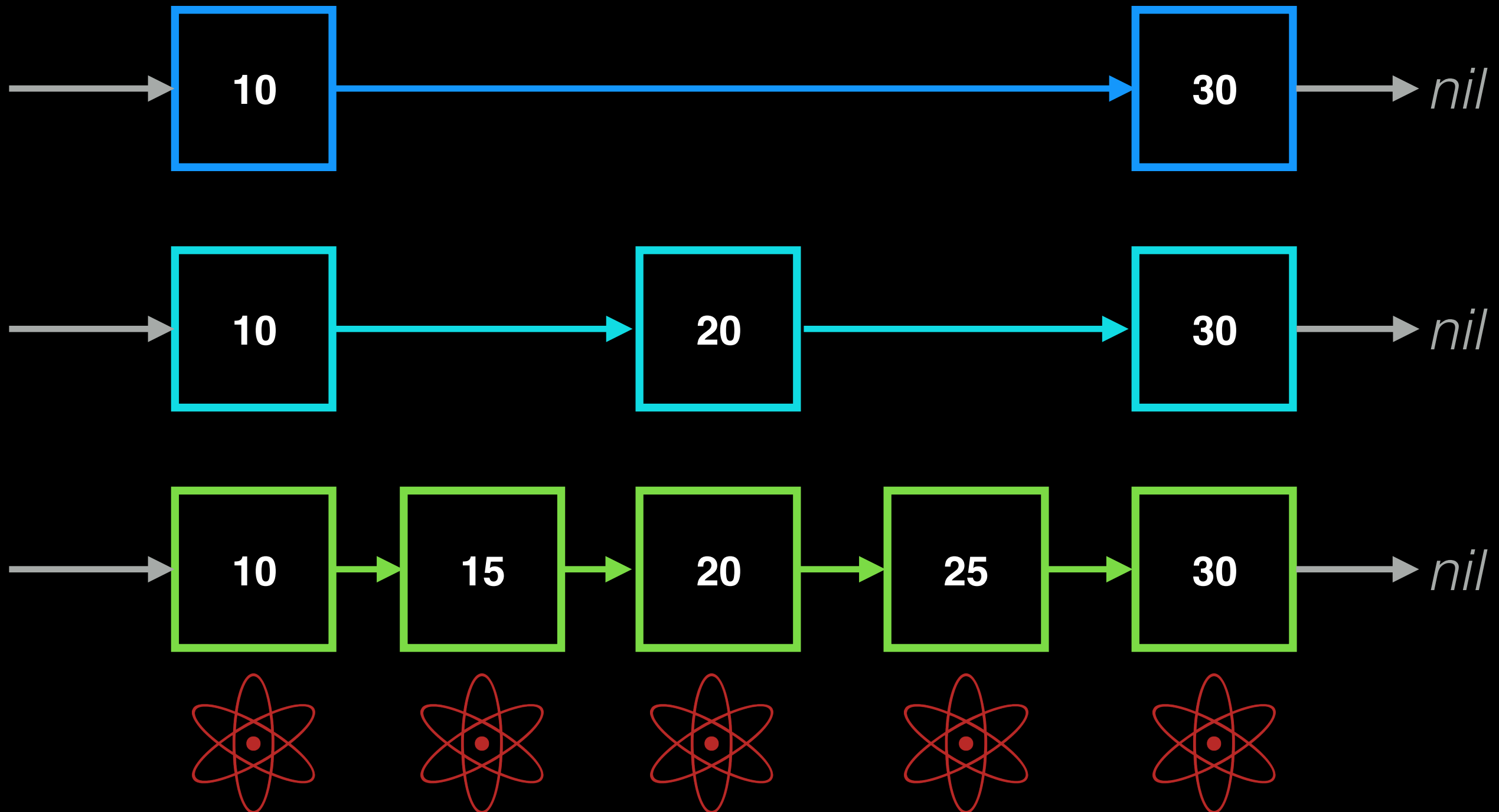
# Concurrency

## Fine-Grained



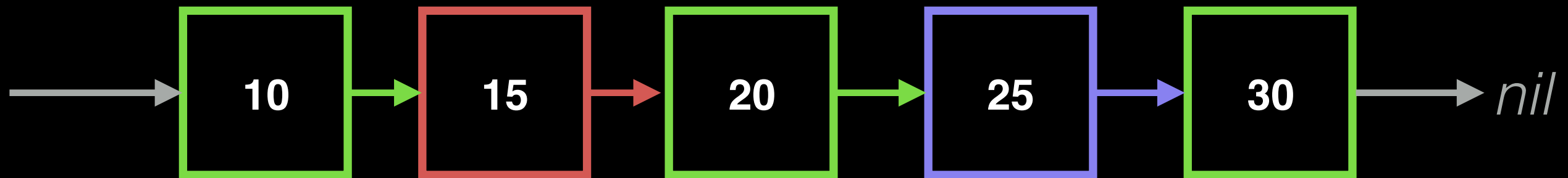
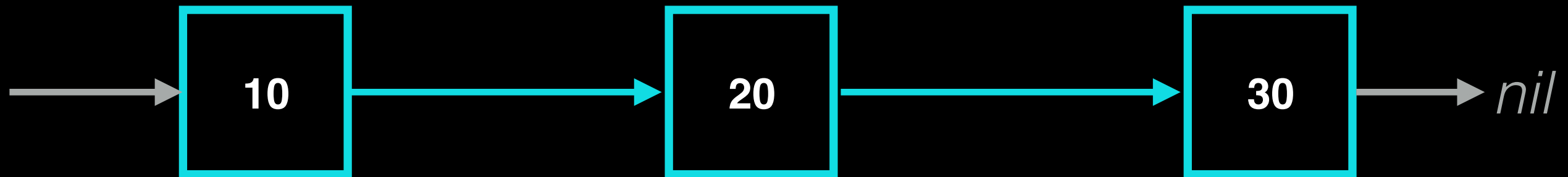
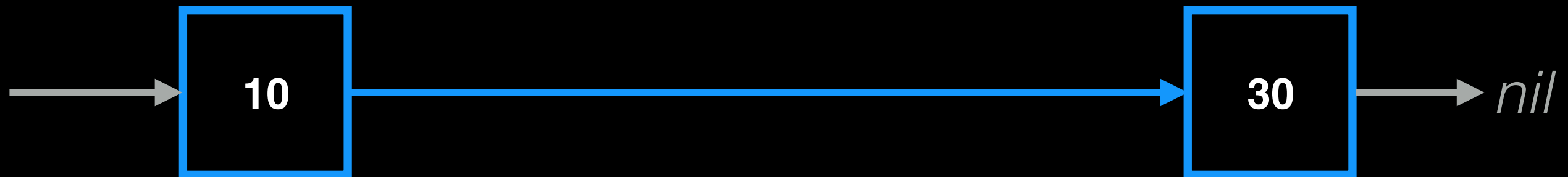
# Concurrency

## Lock-Free



# Concurrency

## Concurrent Modifications



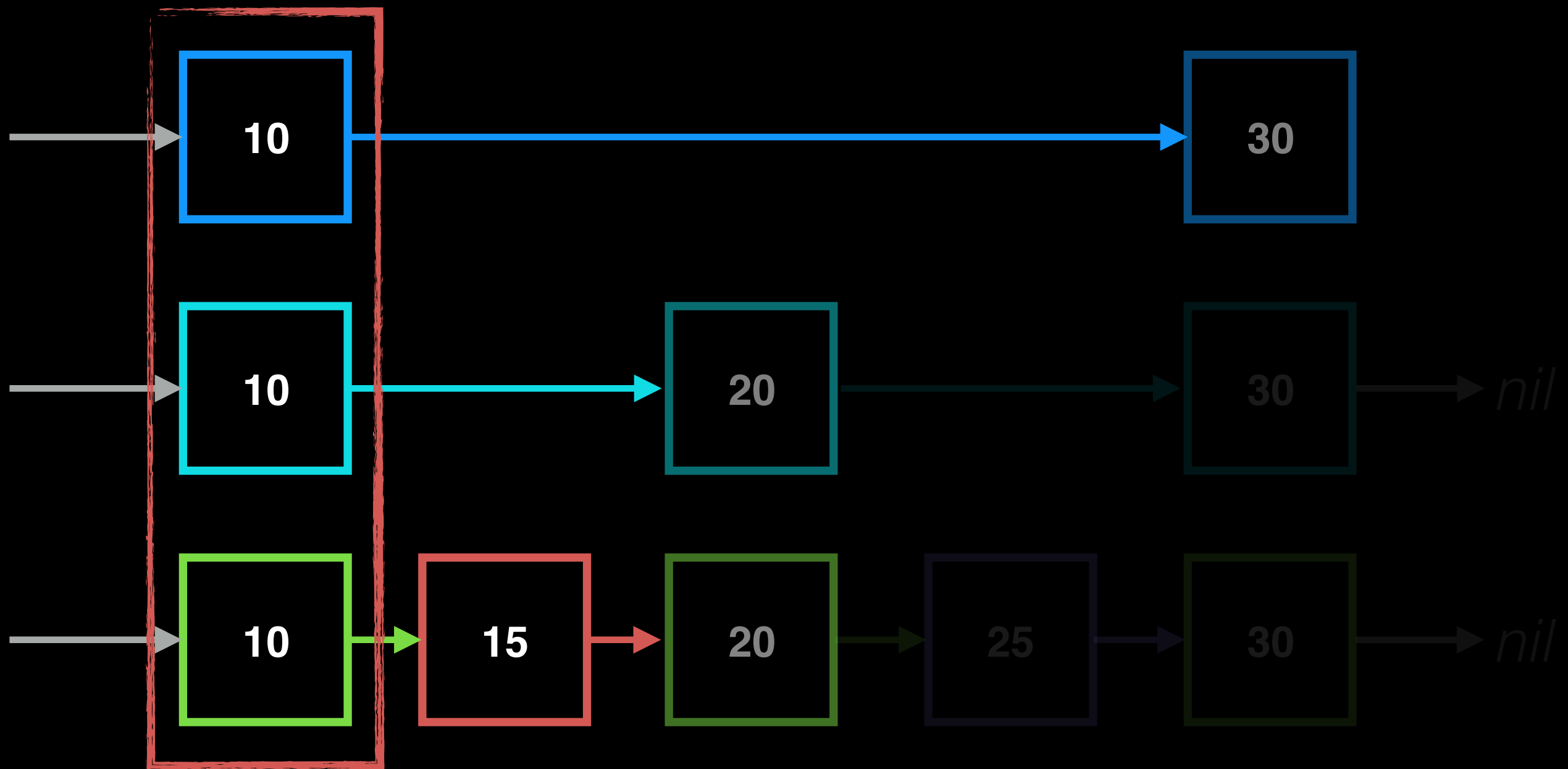
Remove

Insert



# Concurrency

## Concurrent Modifications

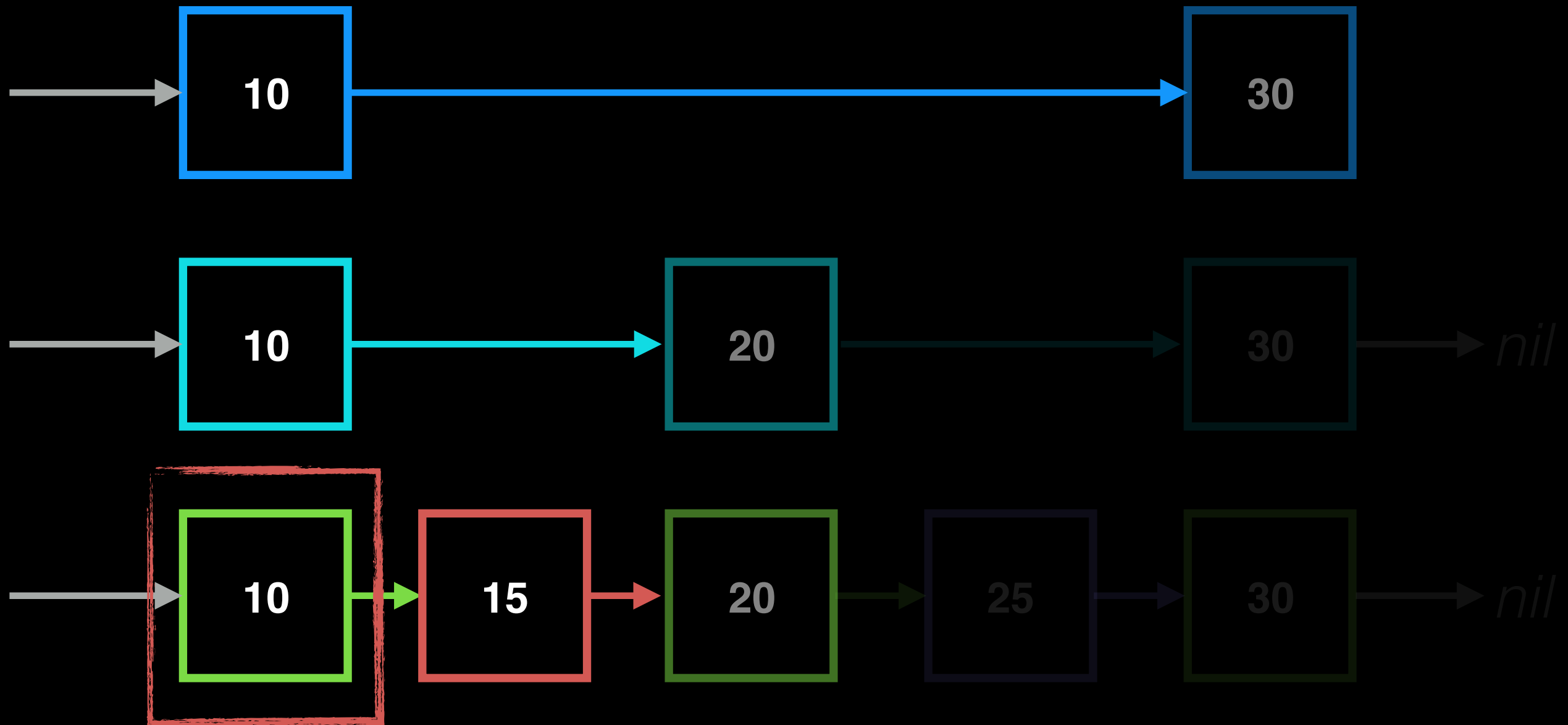


Remove

Insert

# Concurrency

## Concurrent Modifications

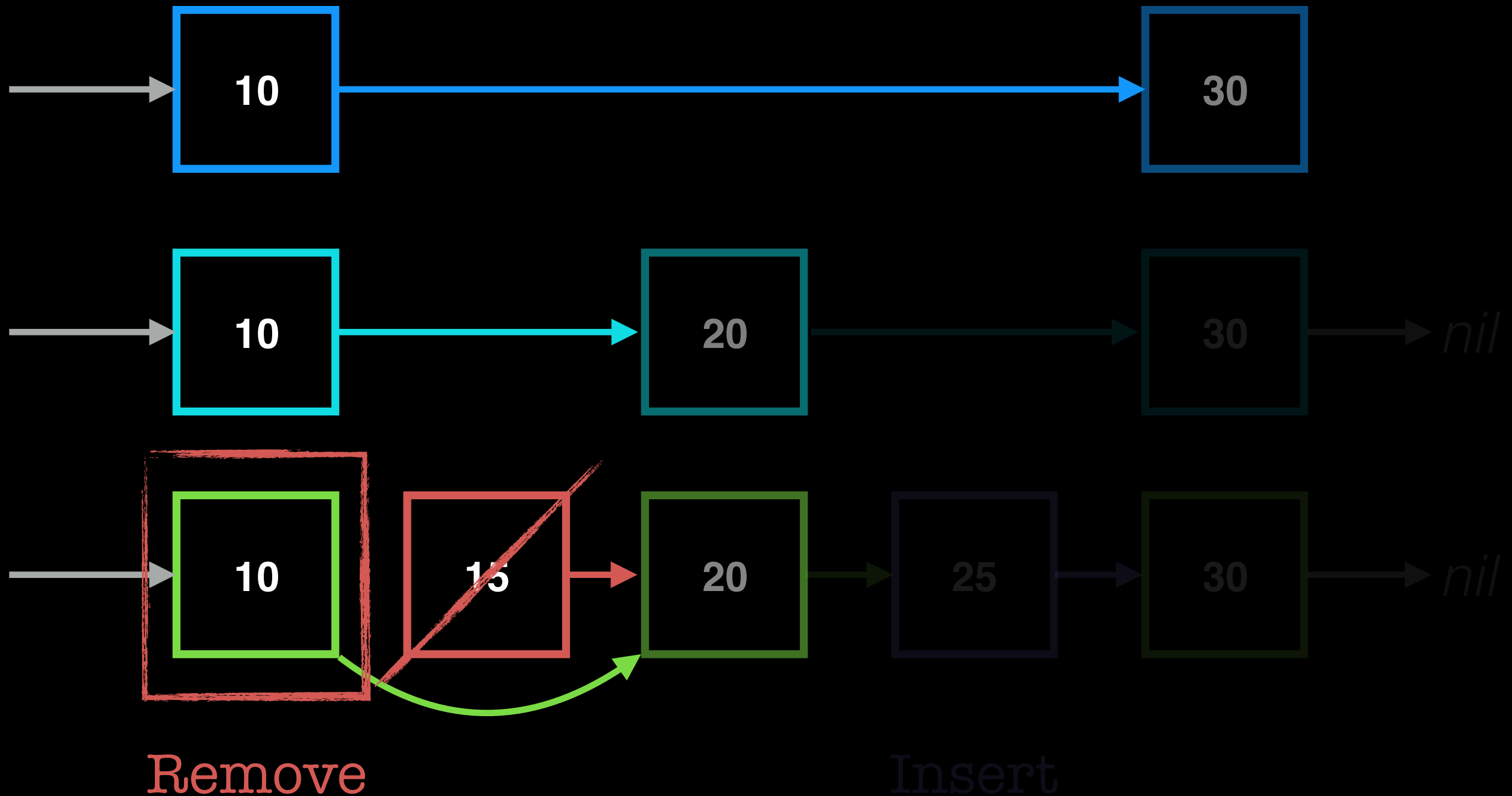


Remove

Insert

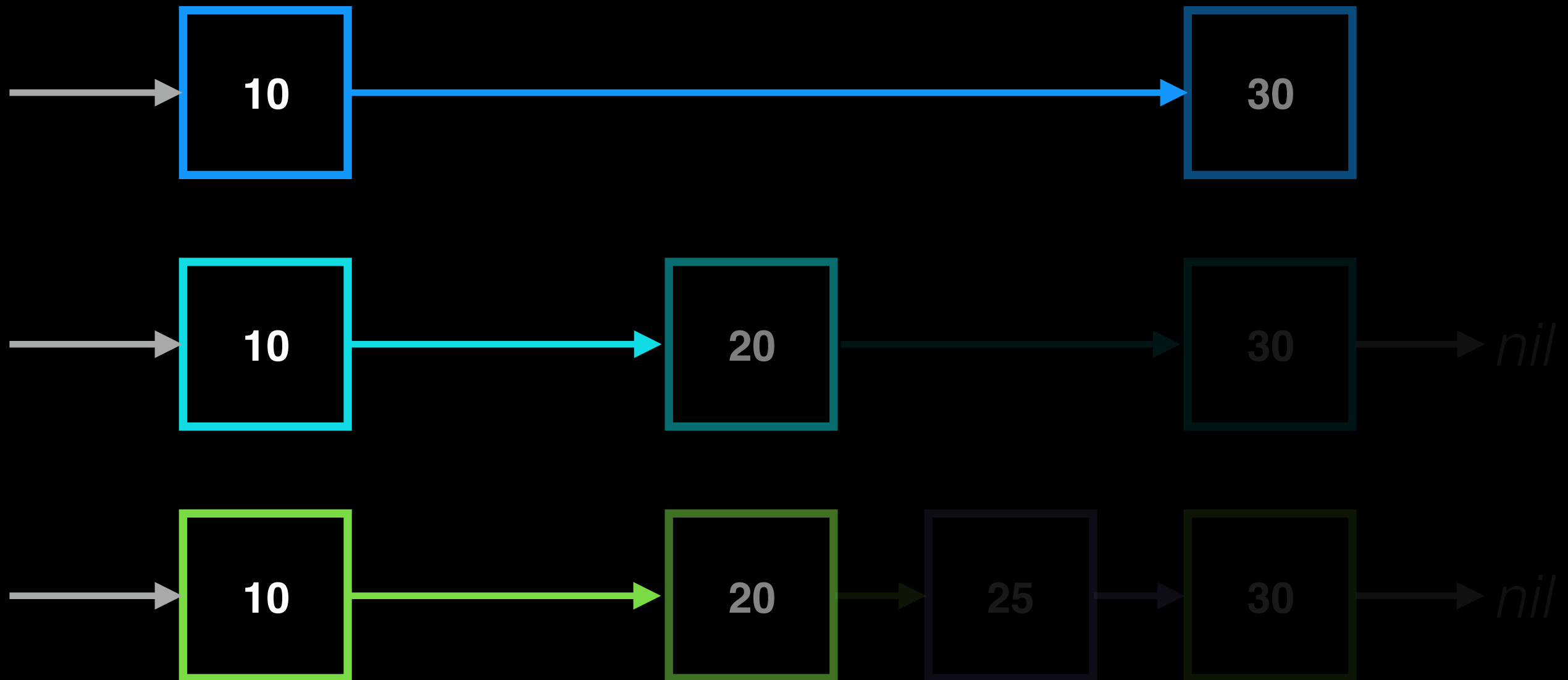
# Concurrency

## Concurrent Modifications



# Concurrency

## Concurrent Modifications

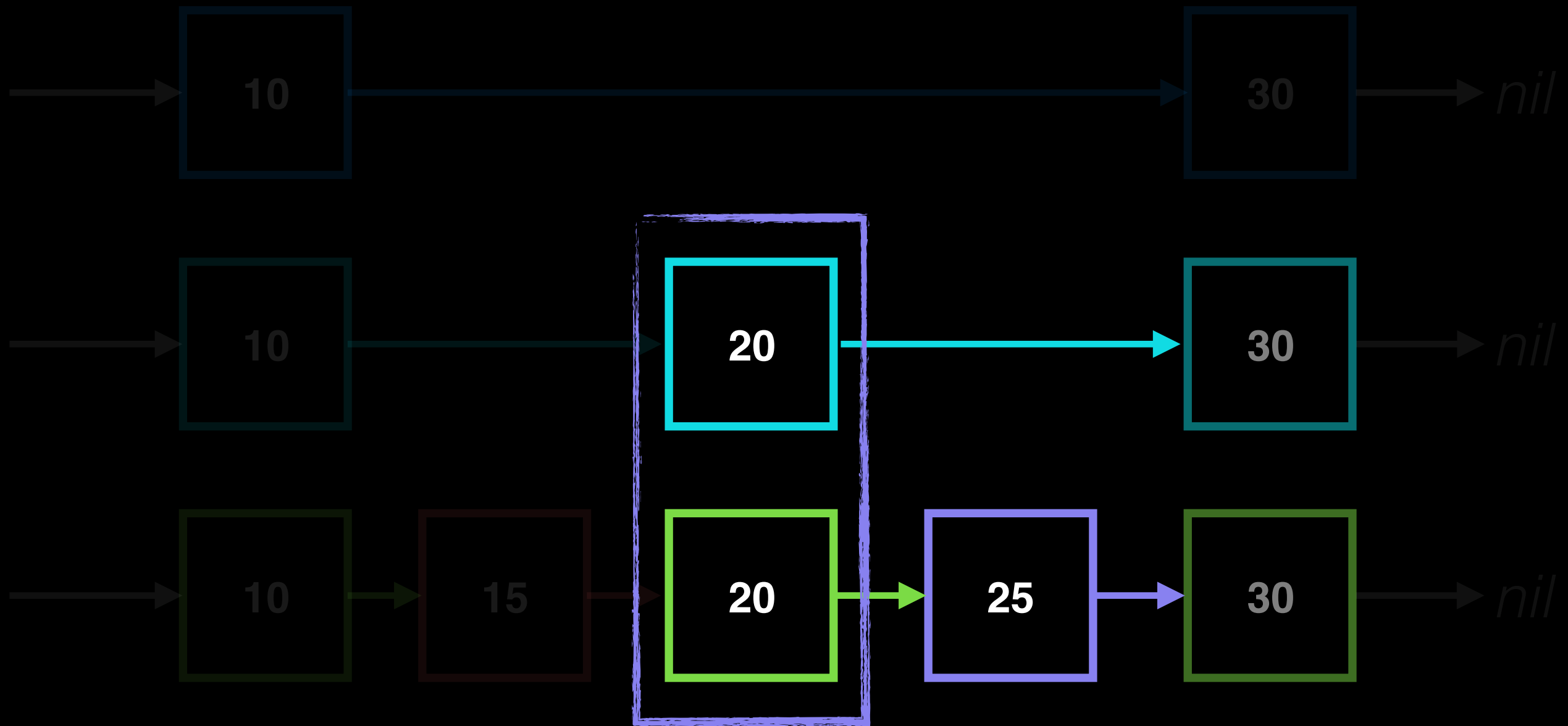


Remove

Insert

# Concurrency

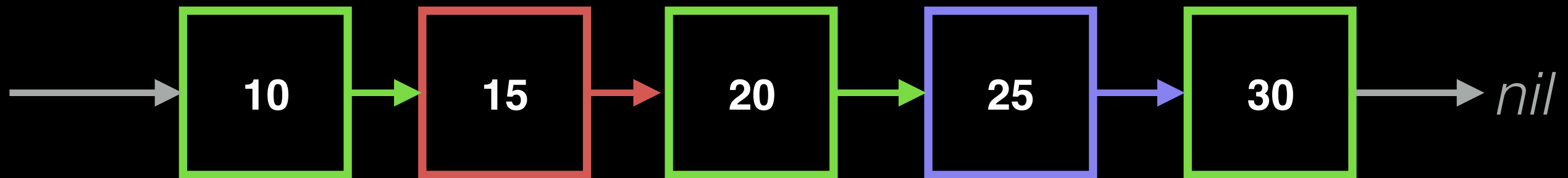
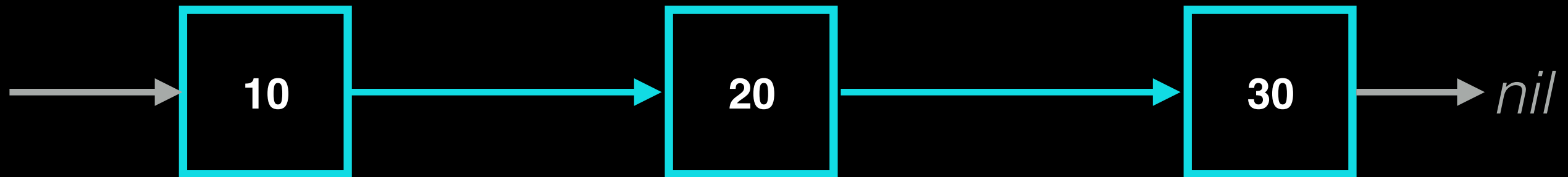
## Concurrent Modifications



Insert

# Concurrency

## Concurrent Modifications



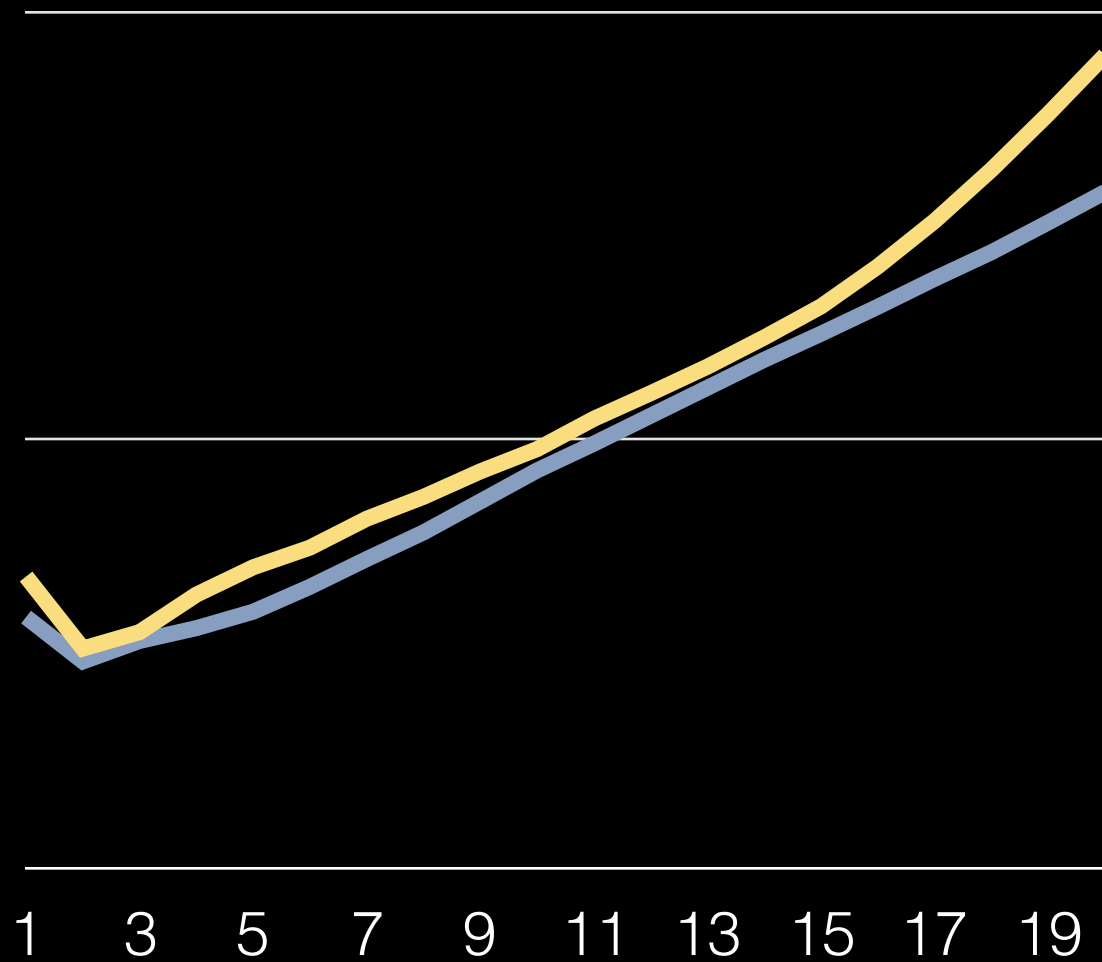
Remove

Insert

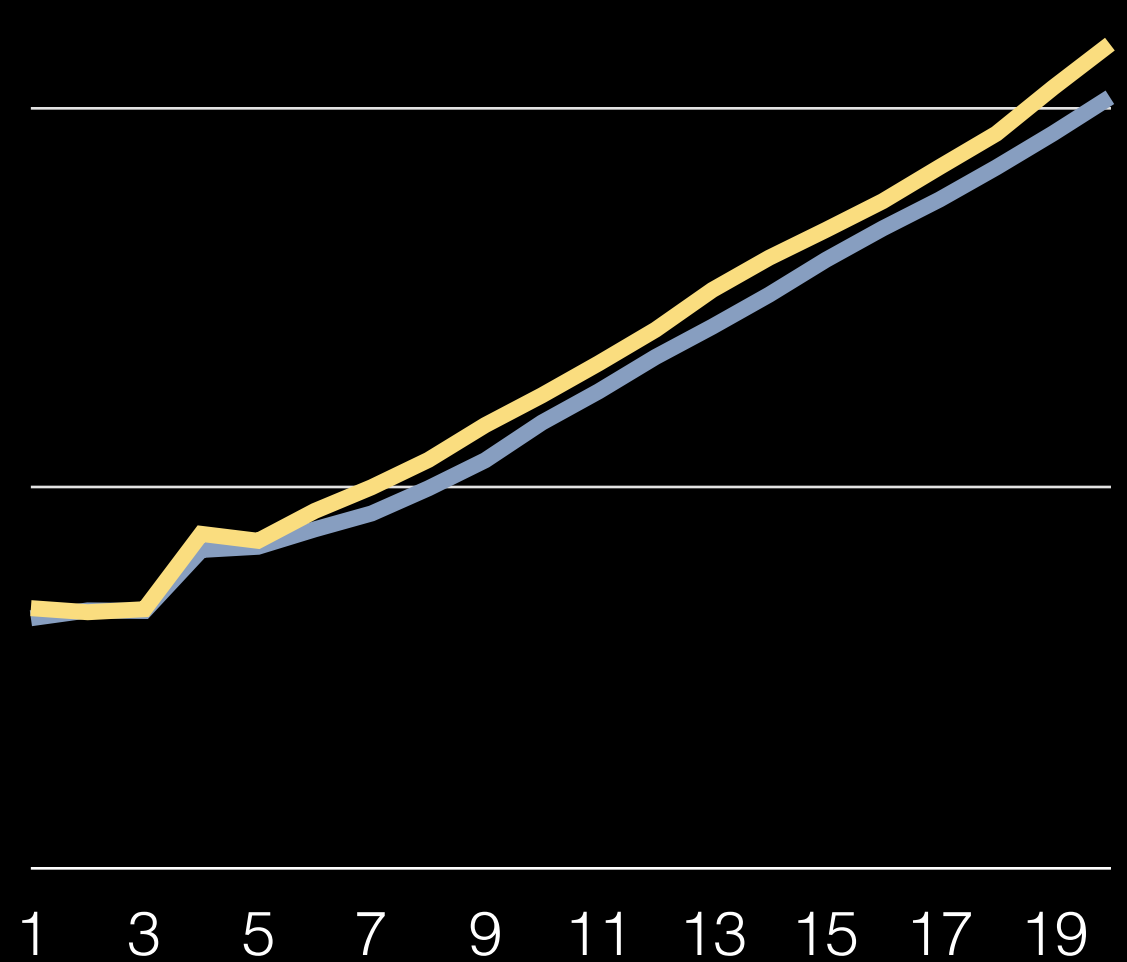
# Insert Performance

$\Theta(\text{Log}(n))$

Fine-Grained Java  
1 Thread



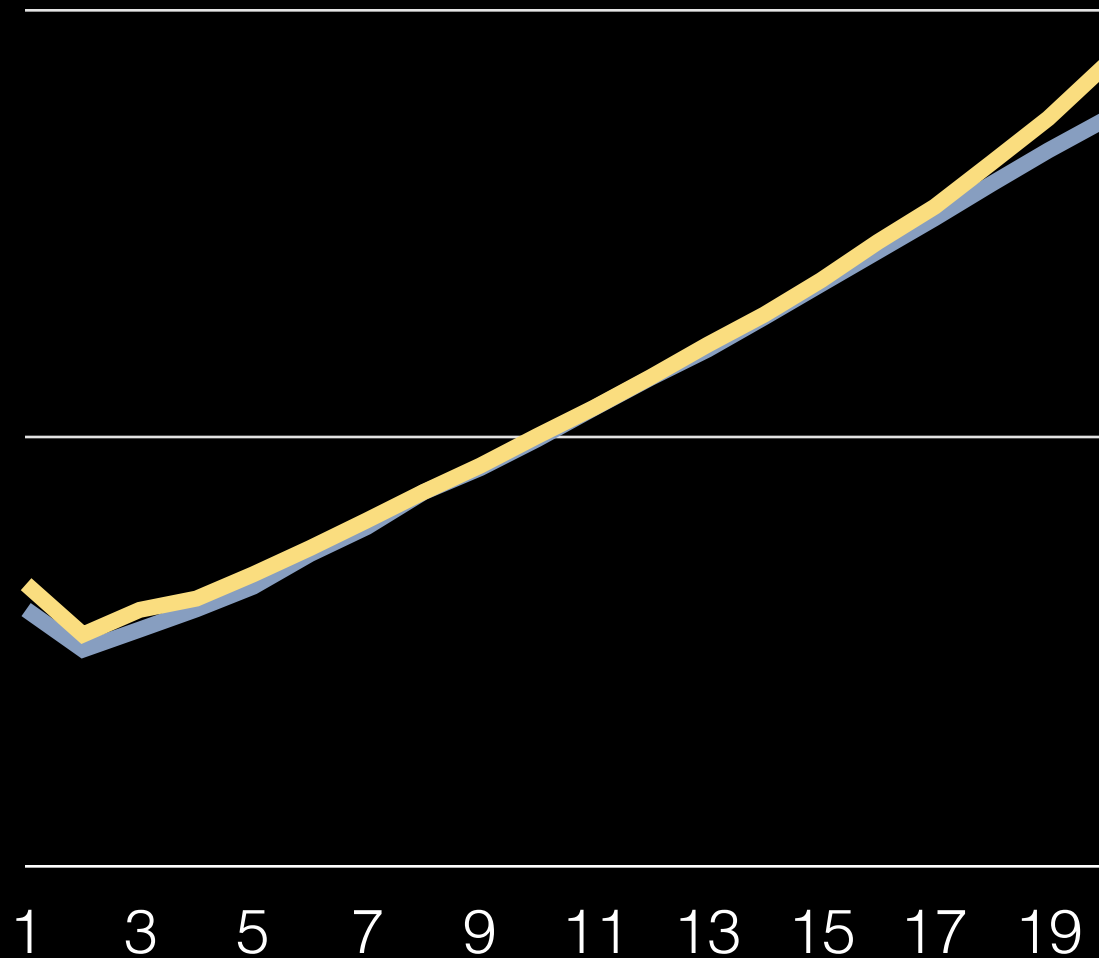
Fine-Grained Java  
8 Threads



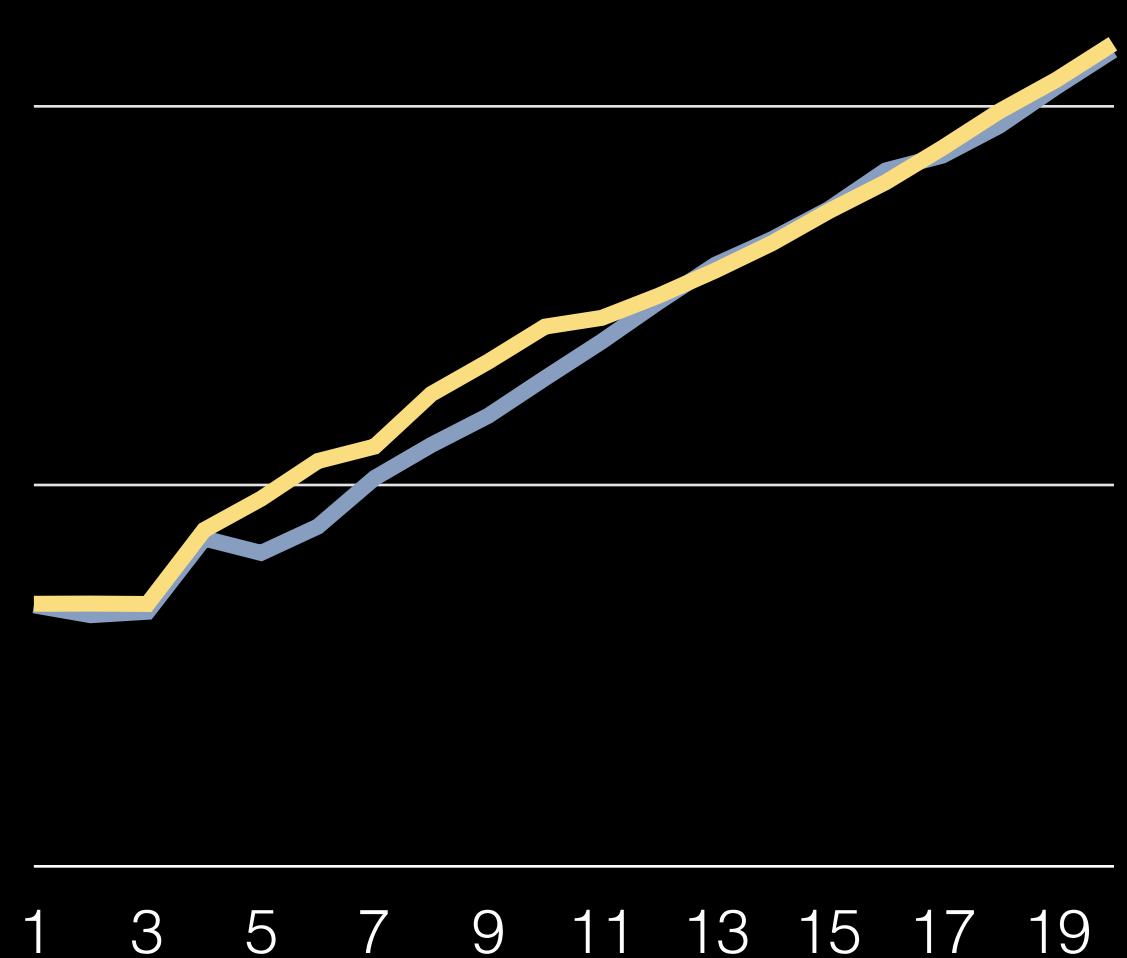
# Remove Performance

$\Theta(\text{Log}(n))$

Fine-Grained Java  
1 Thread



Fine-Grained Java  
8 Threads

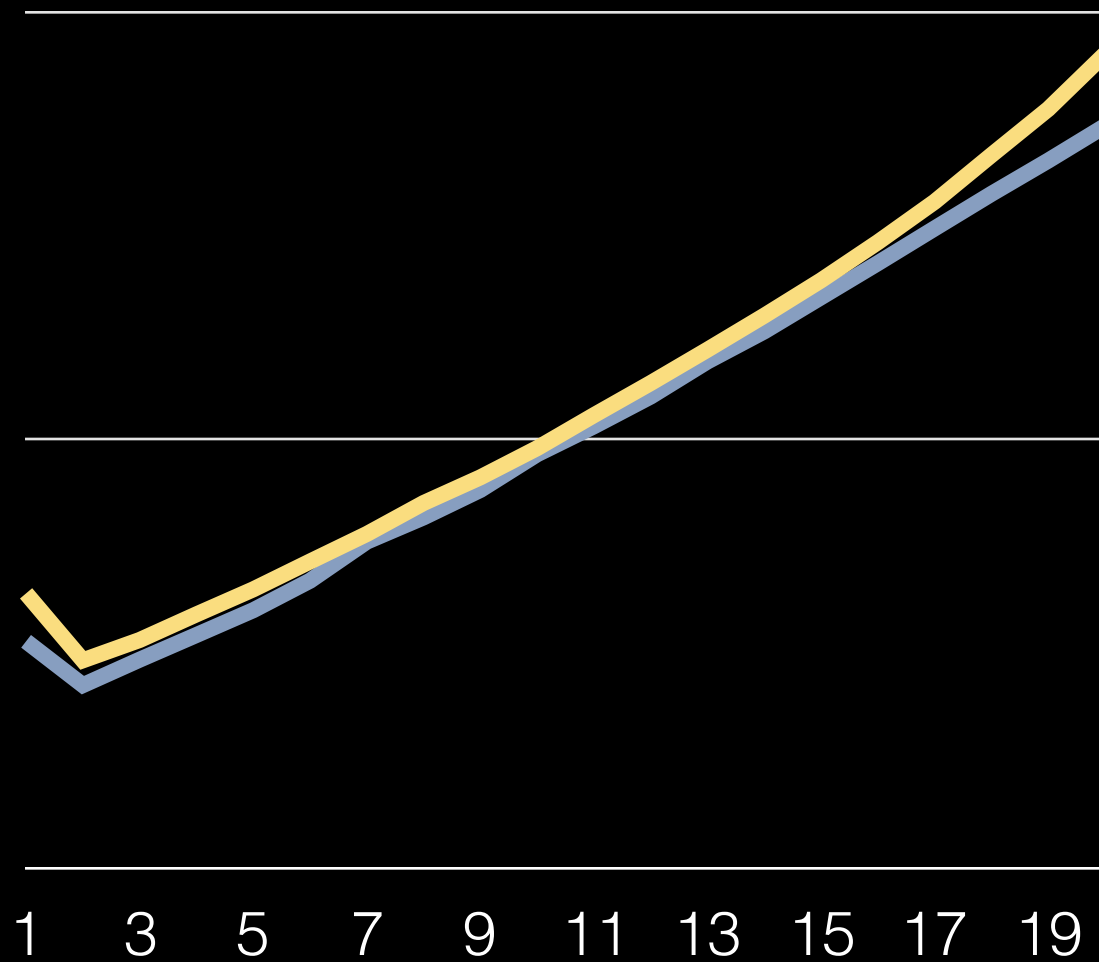




# Search Performance

$$\Theta(\log(n))$$

Fine-Grained Java  
1 Thread



Fine-Grained Java  
8 Threads

