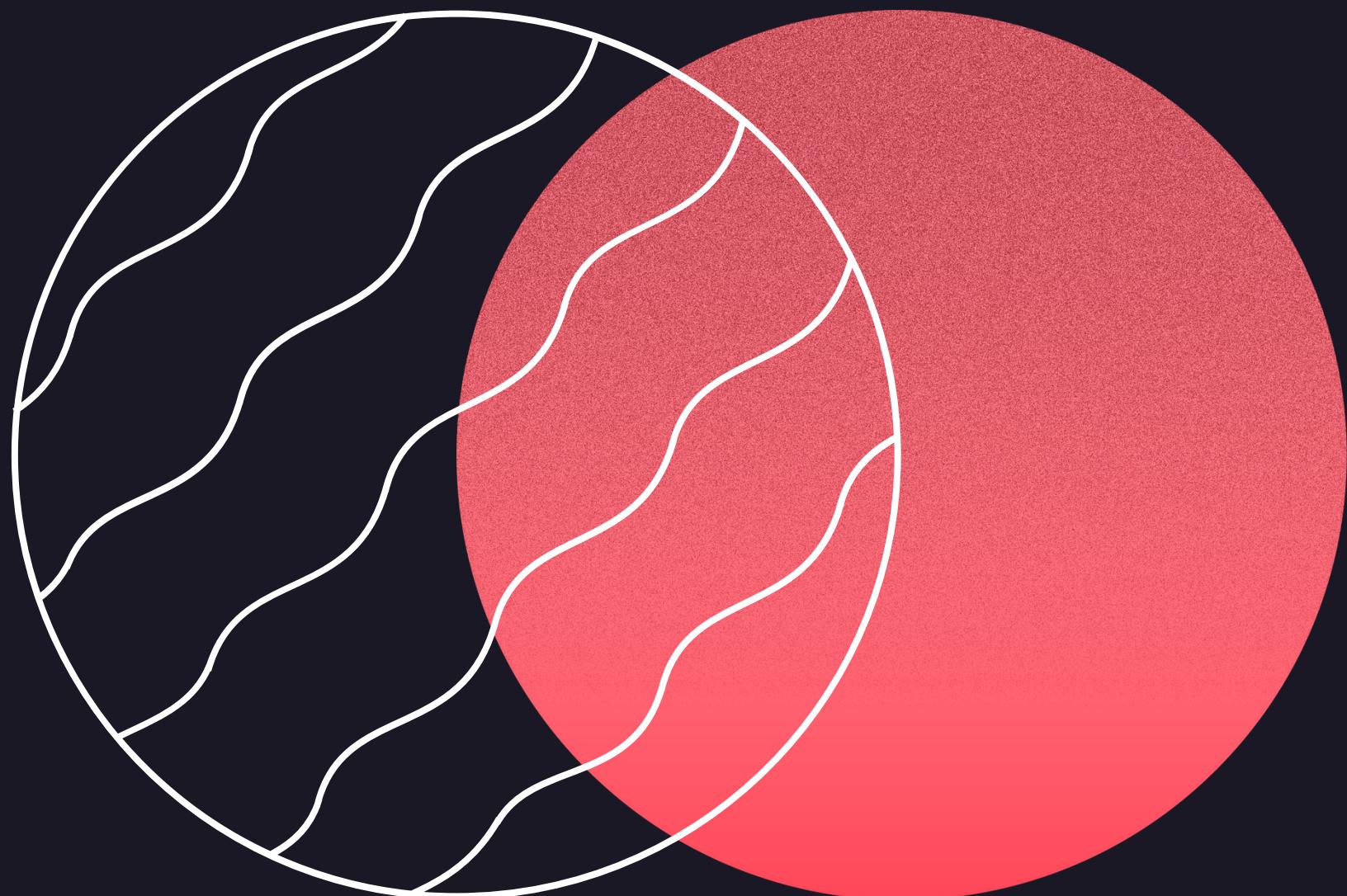


# DS 404

01



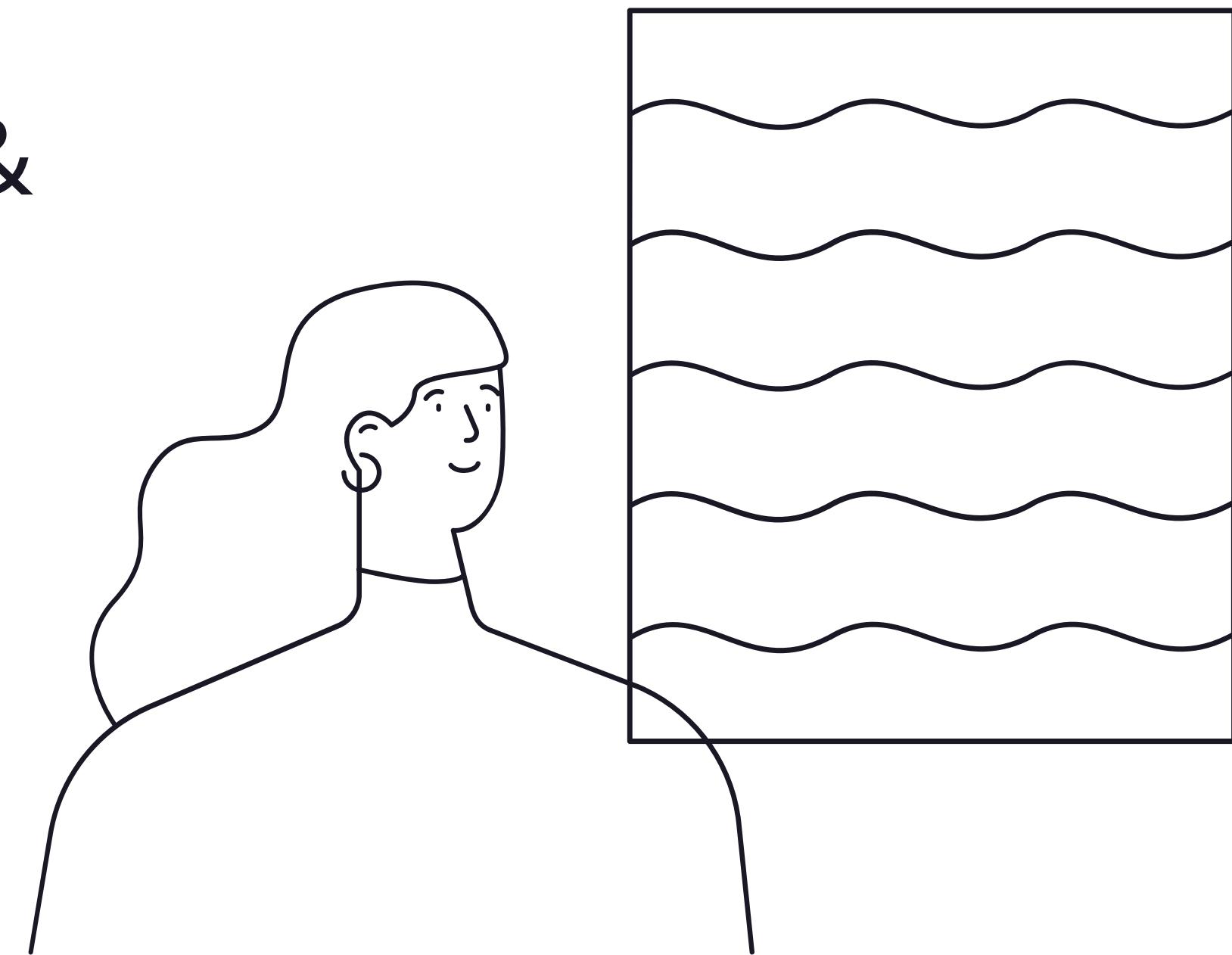
## Let's Talk queues!

Pranav H. Nair

# Refreshing DSA

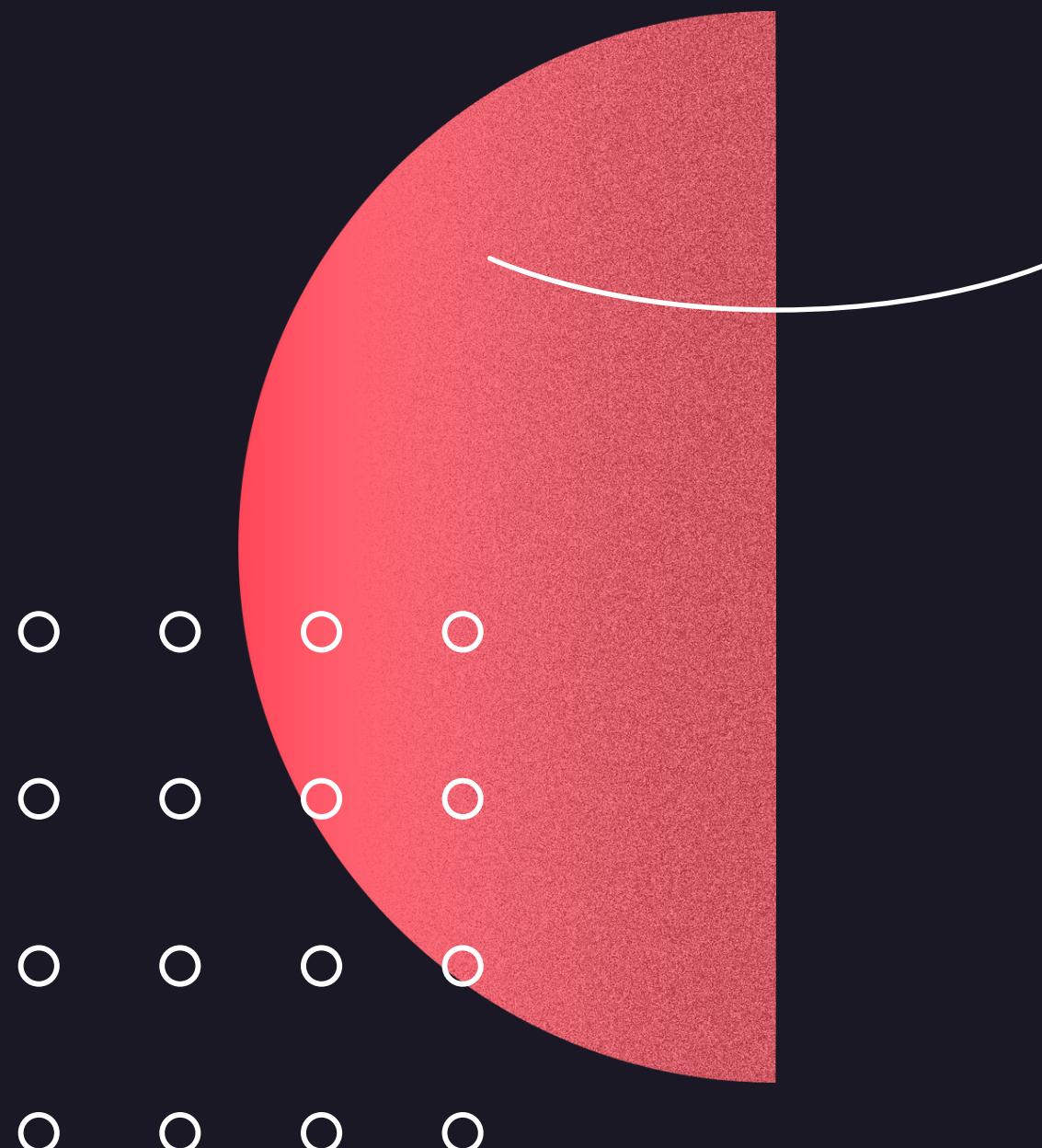
*What we learnt so far*

- Introduction to DSA, Time & Space complexity
- Arrays, Stacks
- Some Simple questions



# What we'll be learning

- What are queues
- How do they work
- Applications



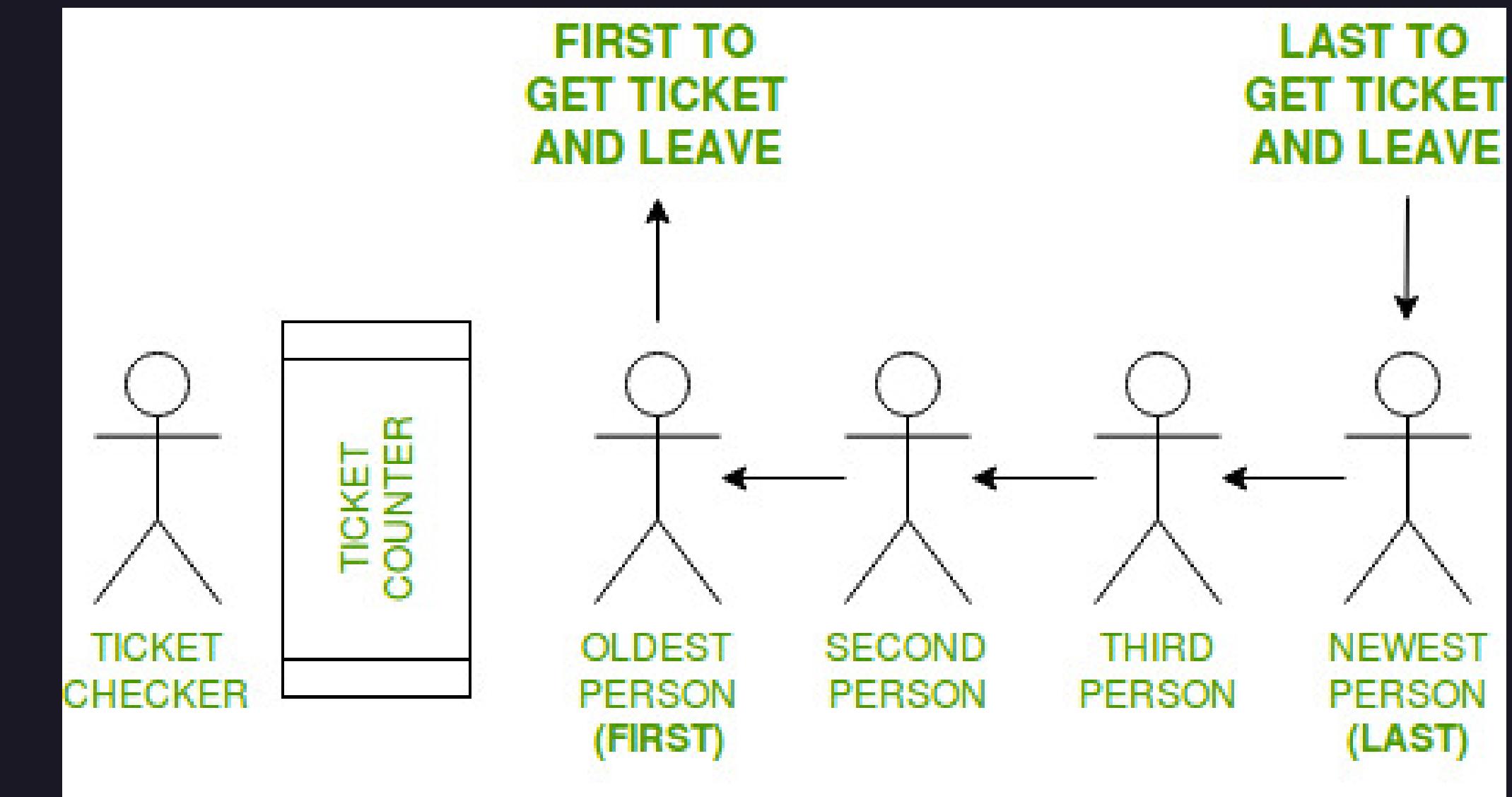
YOU KNOW  
WHAT QUEUES  
ARE!

or do you?

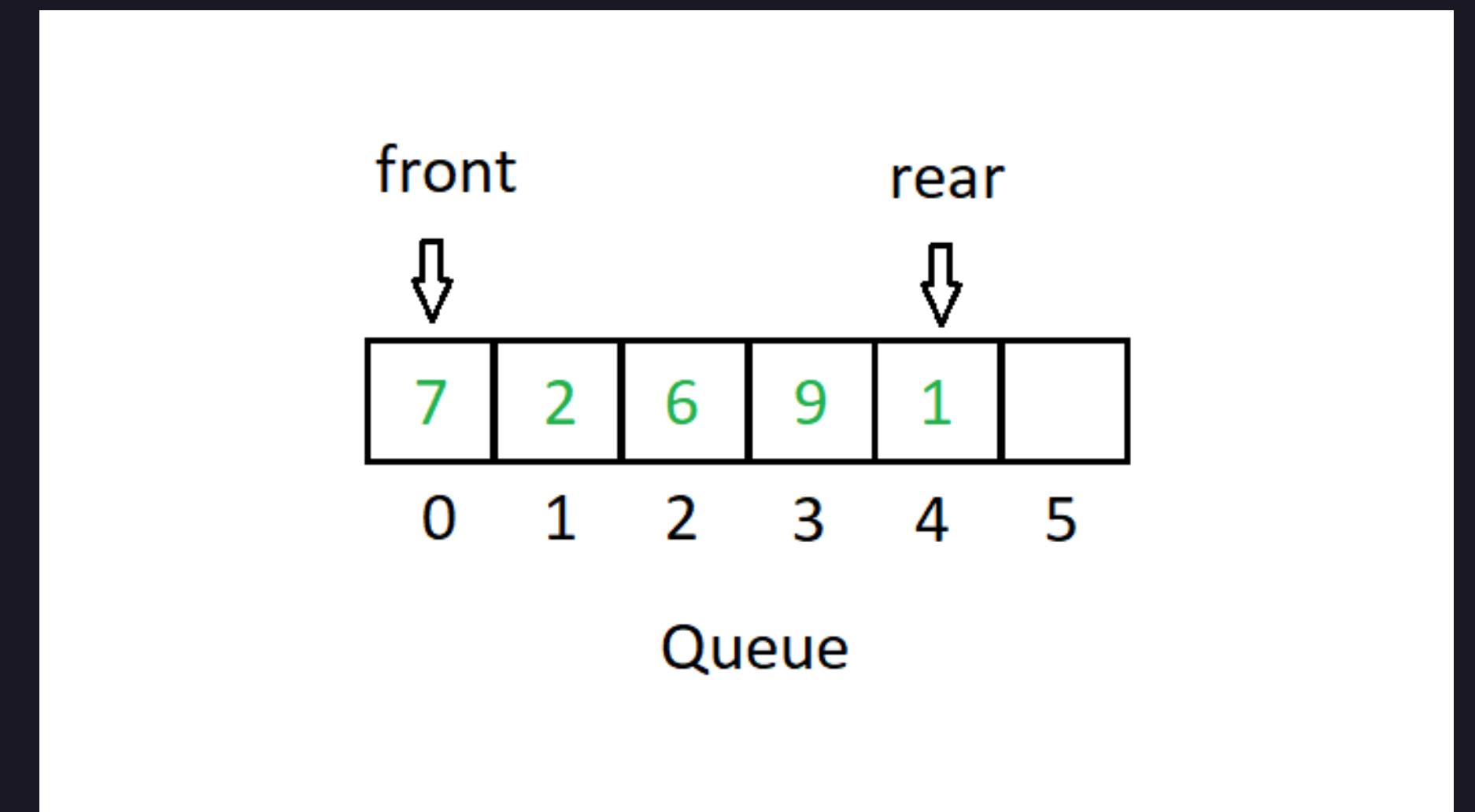
OMG tell me EVERYTHING!



this is  
an *irrl*  
queue!



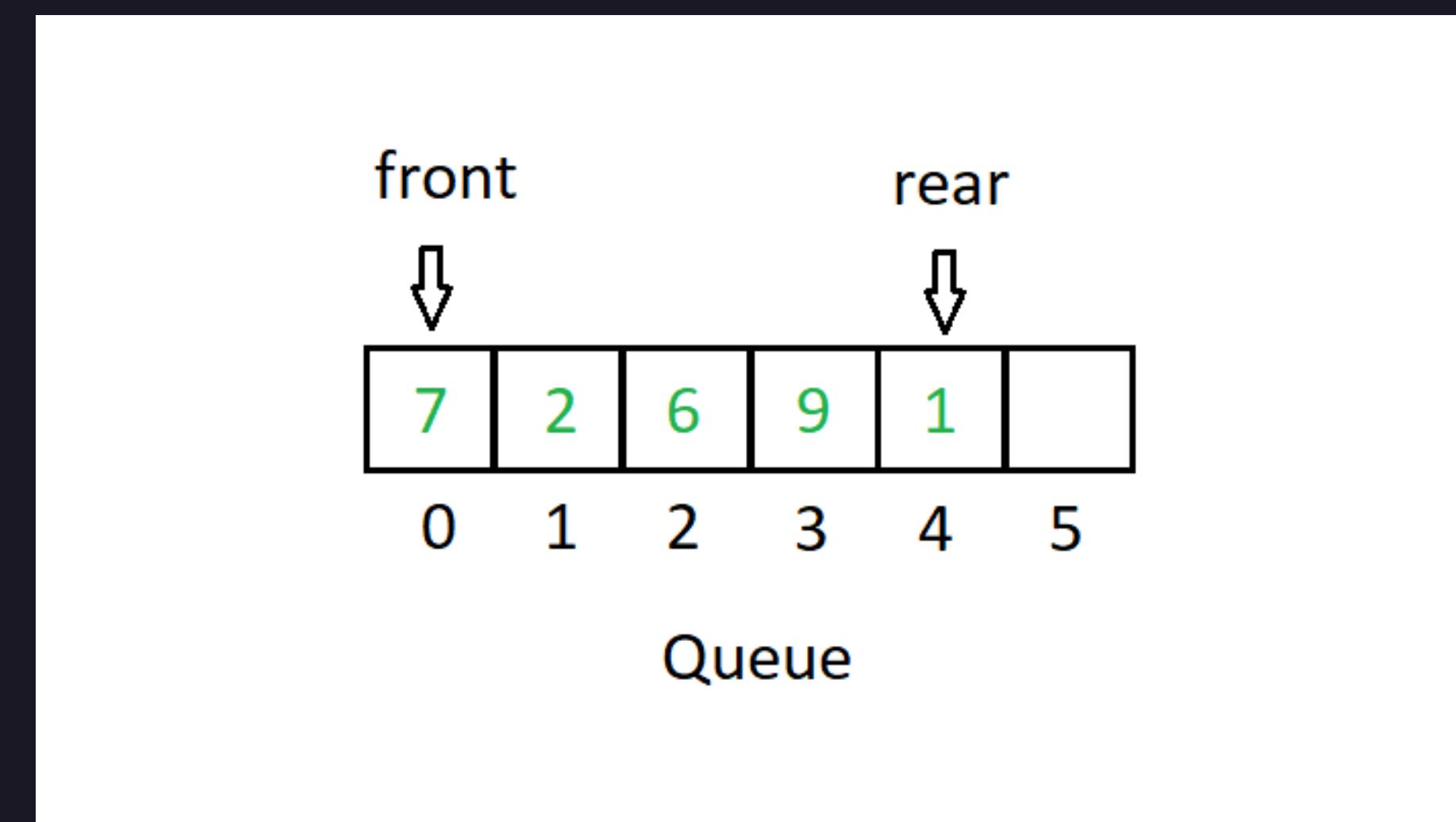
this is  
a  
queue!



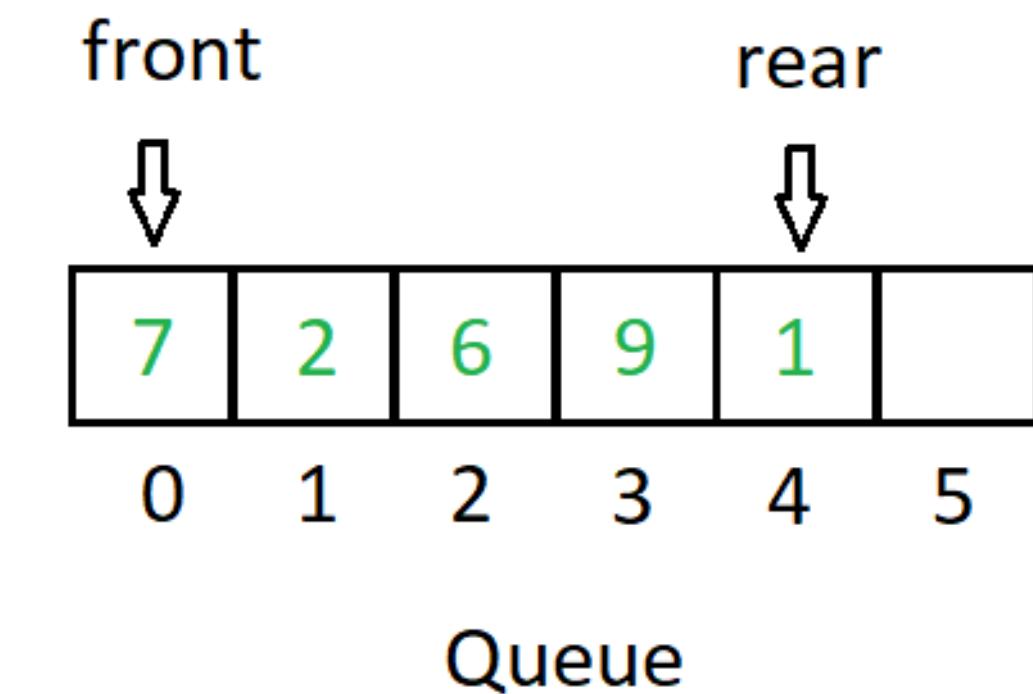
(looks suspiciously like an array)

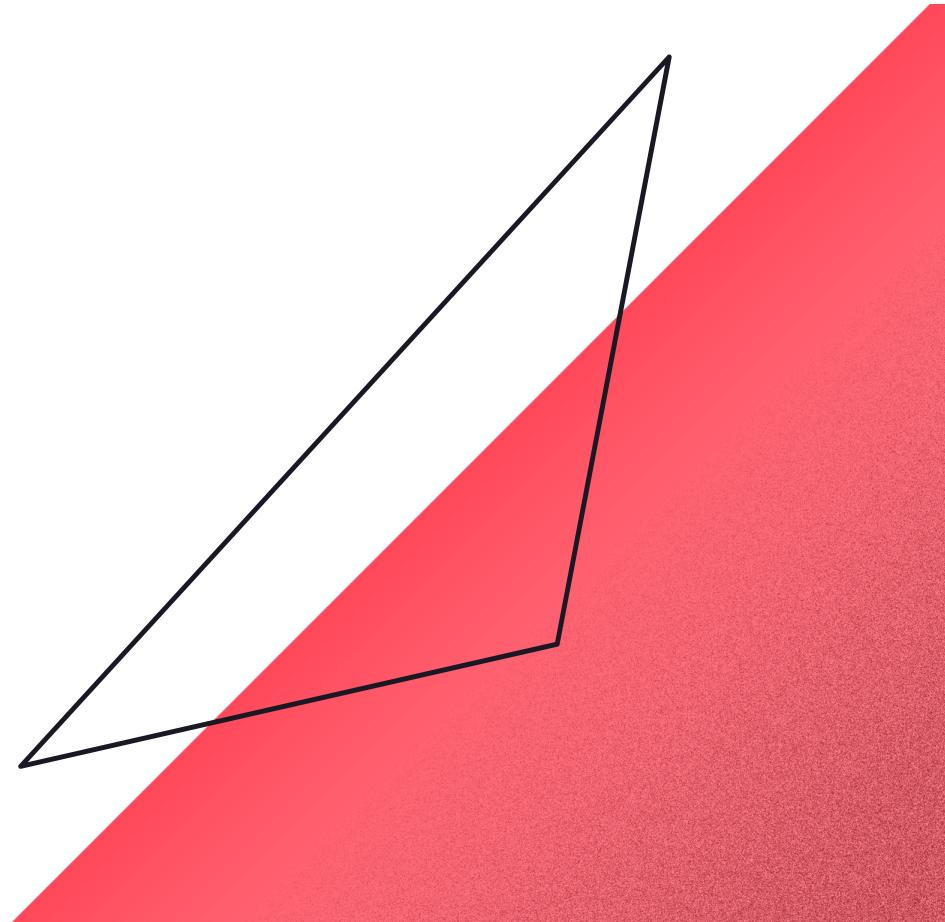
# What is a queue?

A Queue is a linear data structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO)



**There are two  
ends in a queue,  
the front and the  
rear or head and  
the tail**

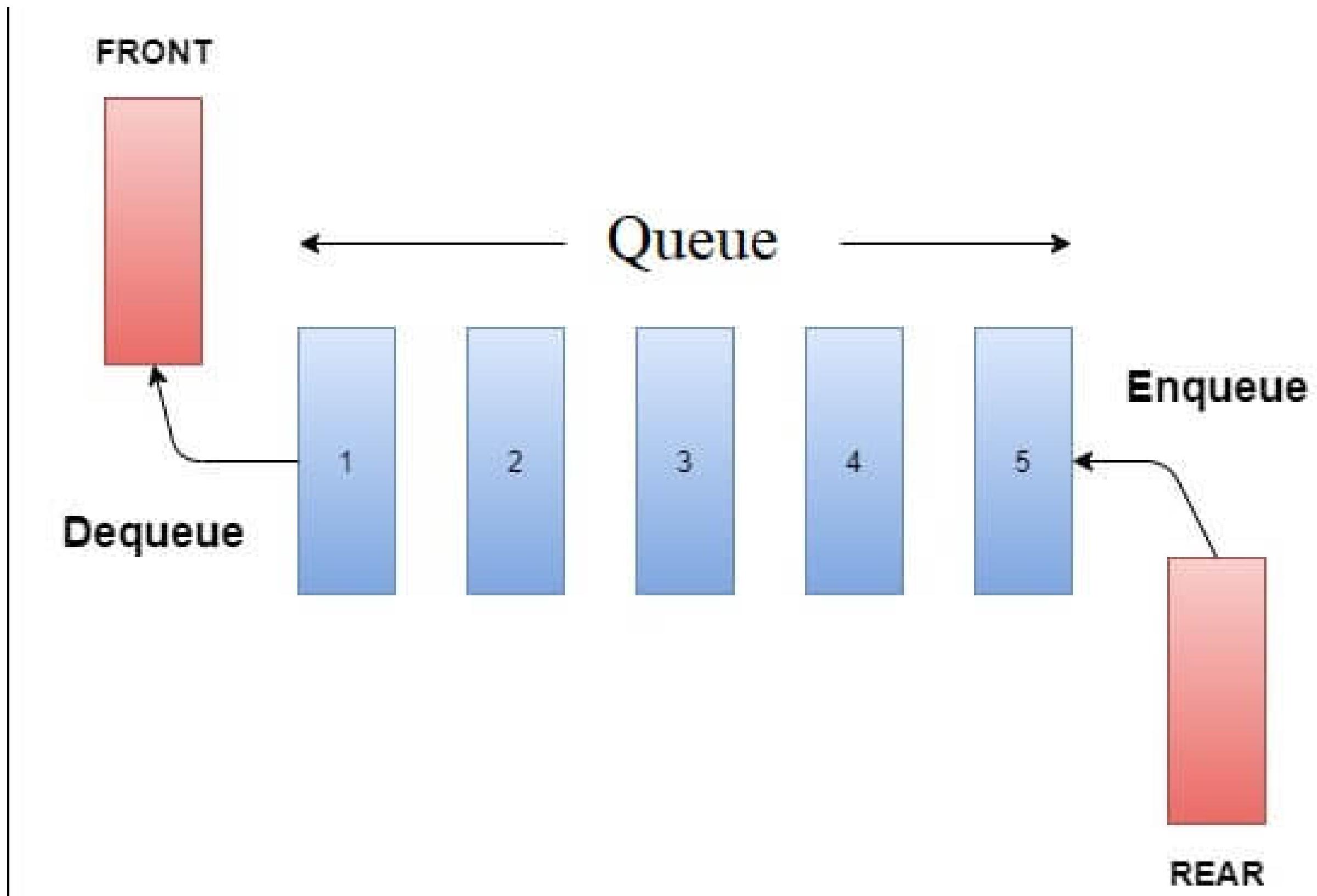


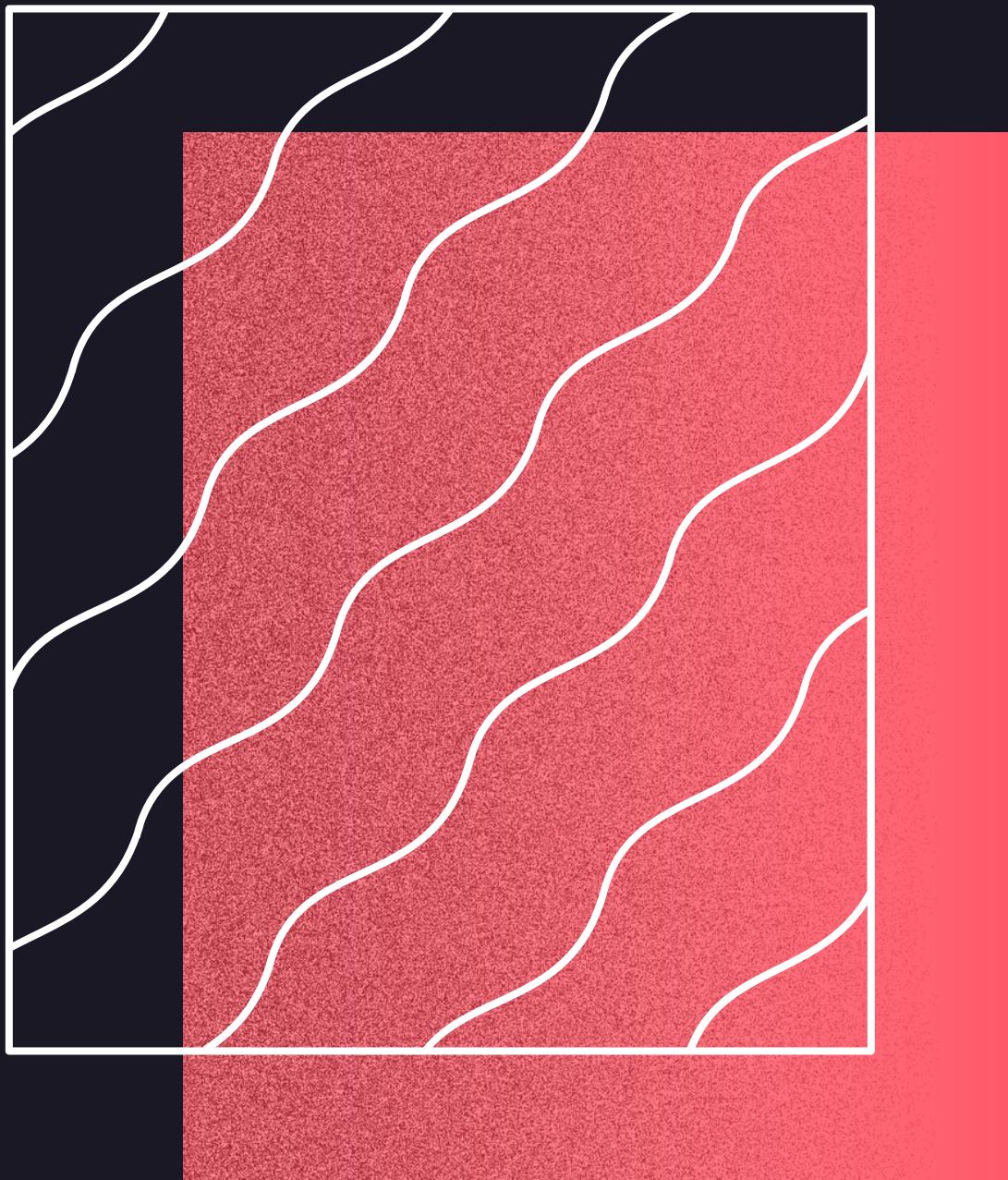


WAIT!

LET'S  
VISUALIZE

# ENTERING ELEMENTS INTO A QUEUE IS CALLED ENQUEUE AND REMOVING ELEMENTS IS CALLED DEQUEUE

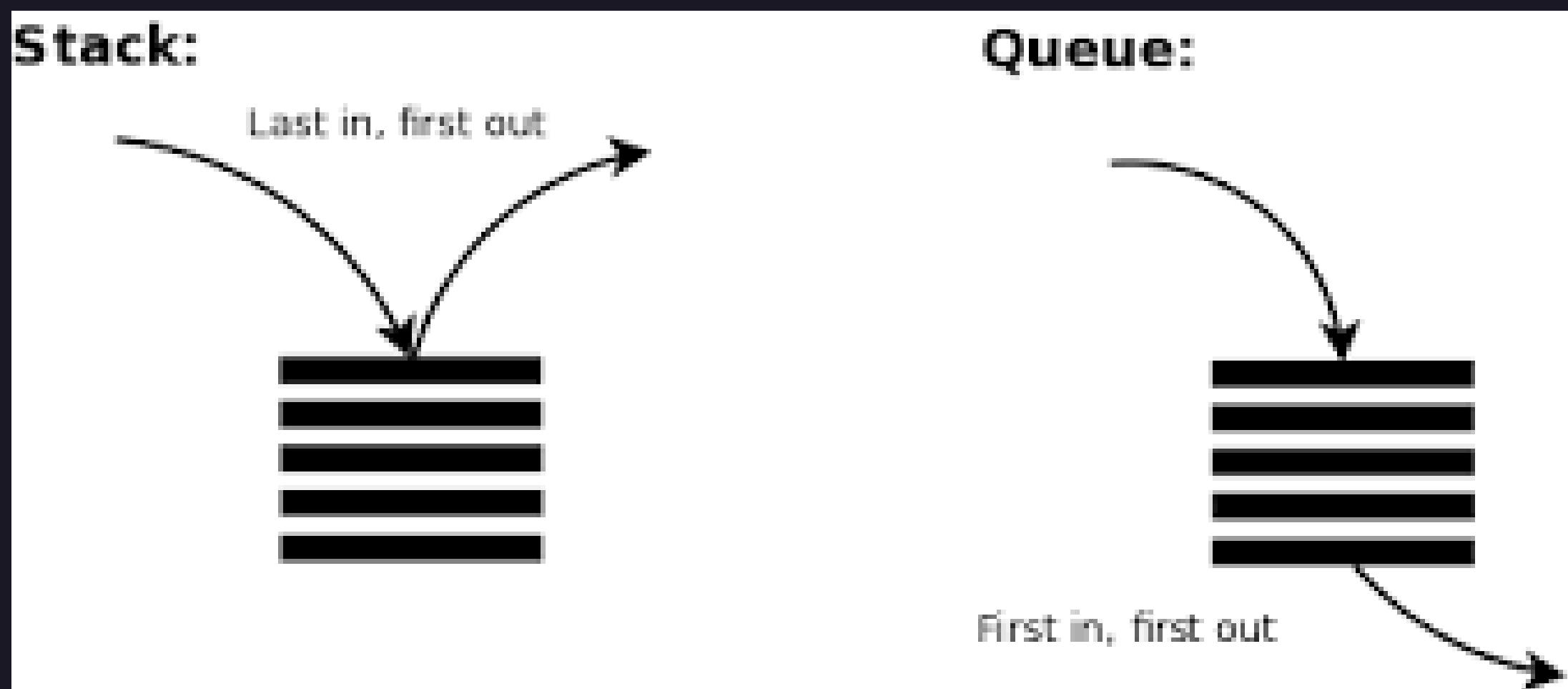




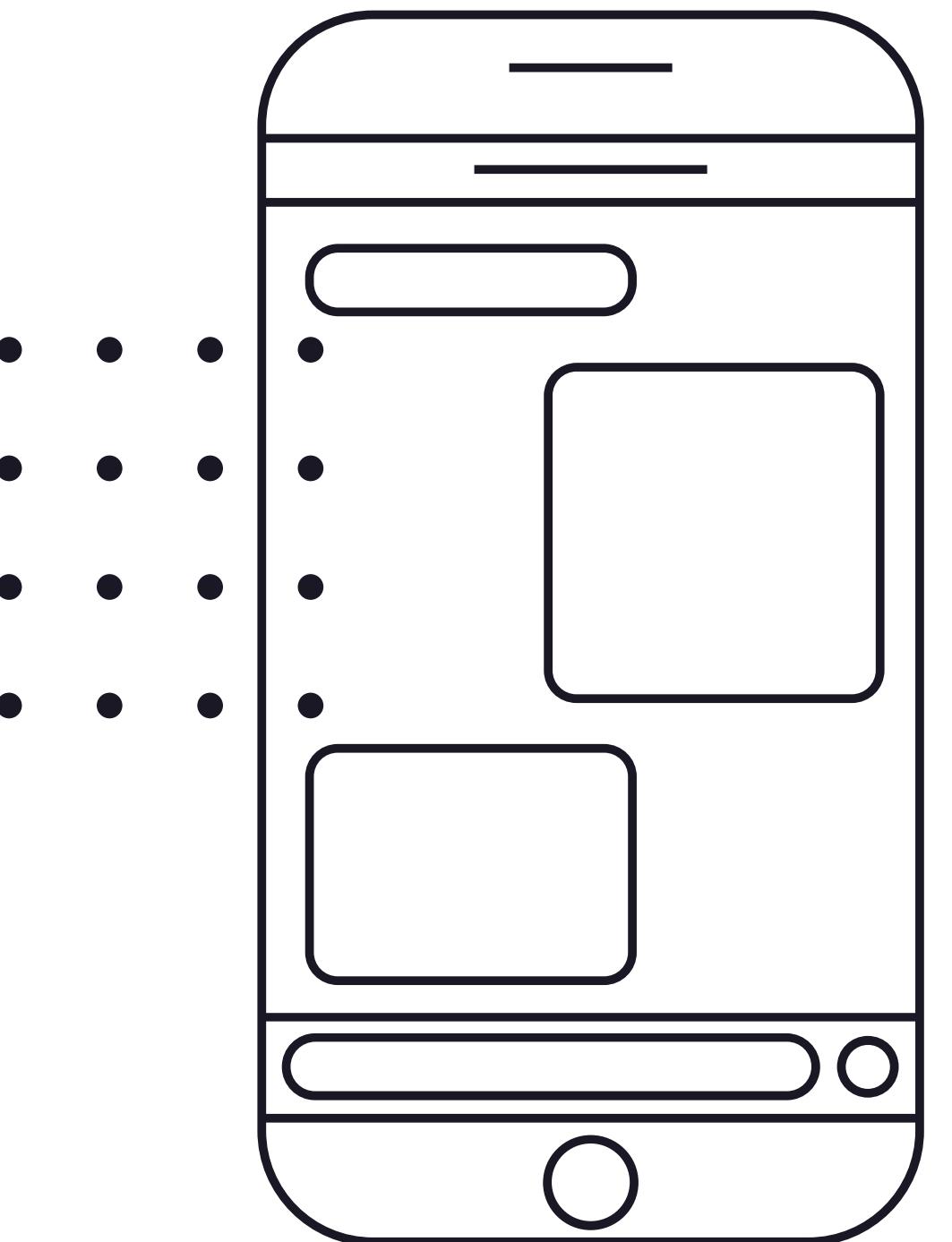
# what's in a name?

WHAT'S THE DIFFERENCE  
BETWEEN A STACK AND  
A QUEUE?

IN A STACK WE REMOVE THE ITEM THE MOST RECENTLY ADDED; IN A QUEUE, WE REMOVE THE ITEM THE LEAST RECENTLY ADDED<sup>67</sup>



# LET'S CODE!



# QUESTION 1

## PERFORM THE FOLLOWING OPERATIONS:

Enqueue(1)

Enqueue(3)

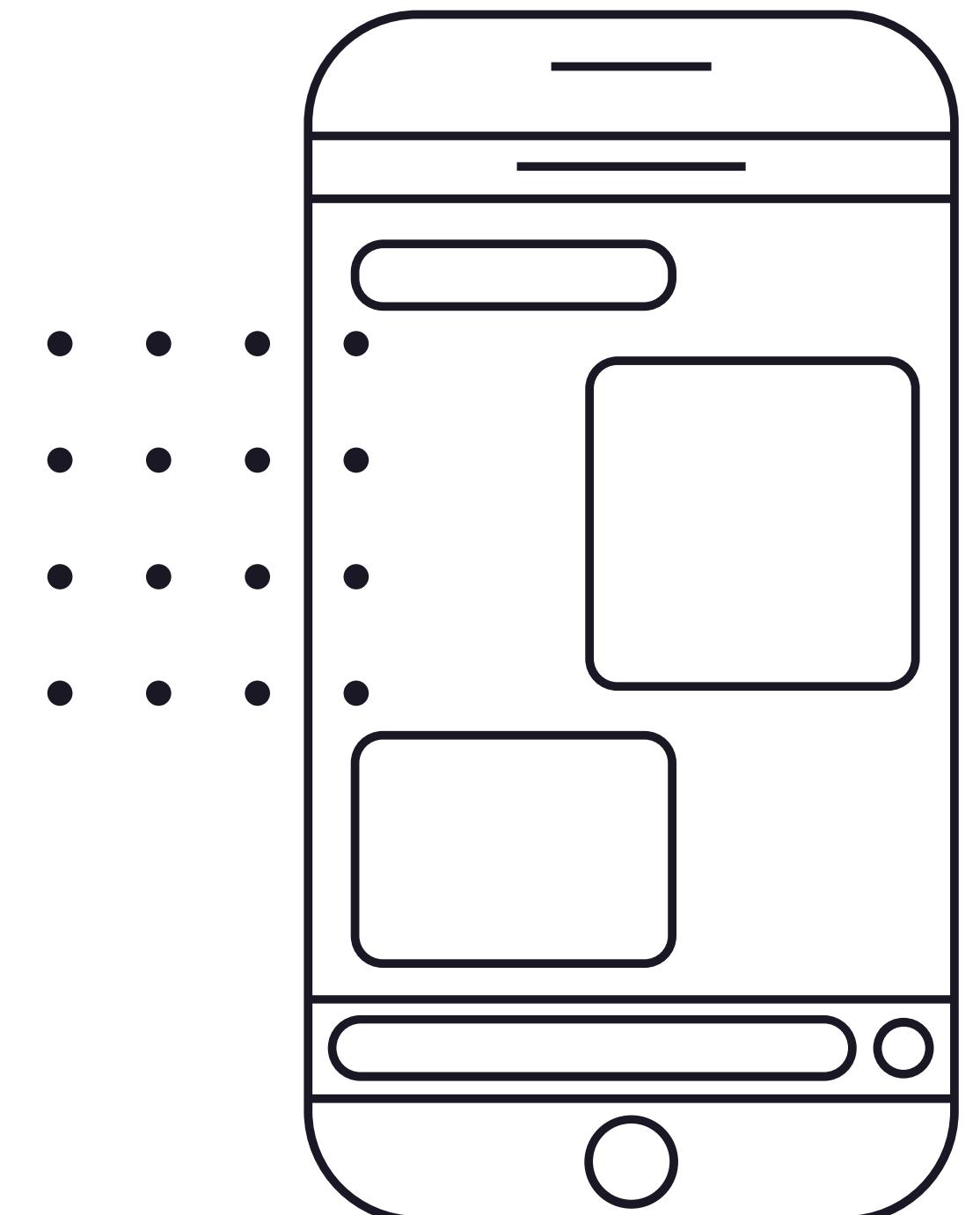
Enqueue(2)

Enqueue(7)

Dequeue()

Dequeue()

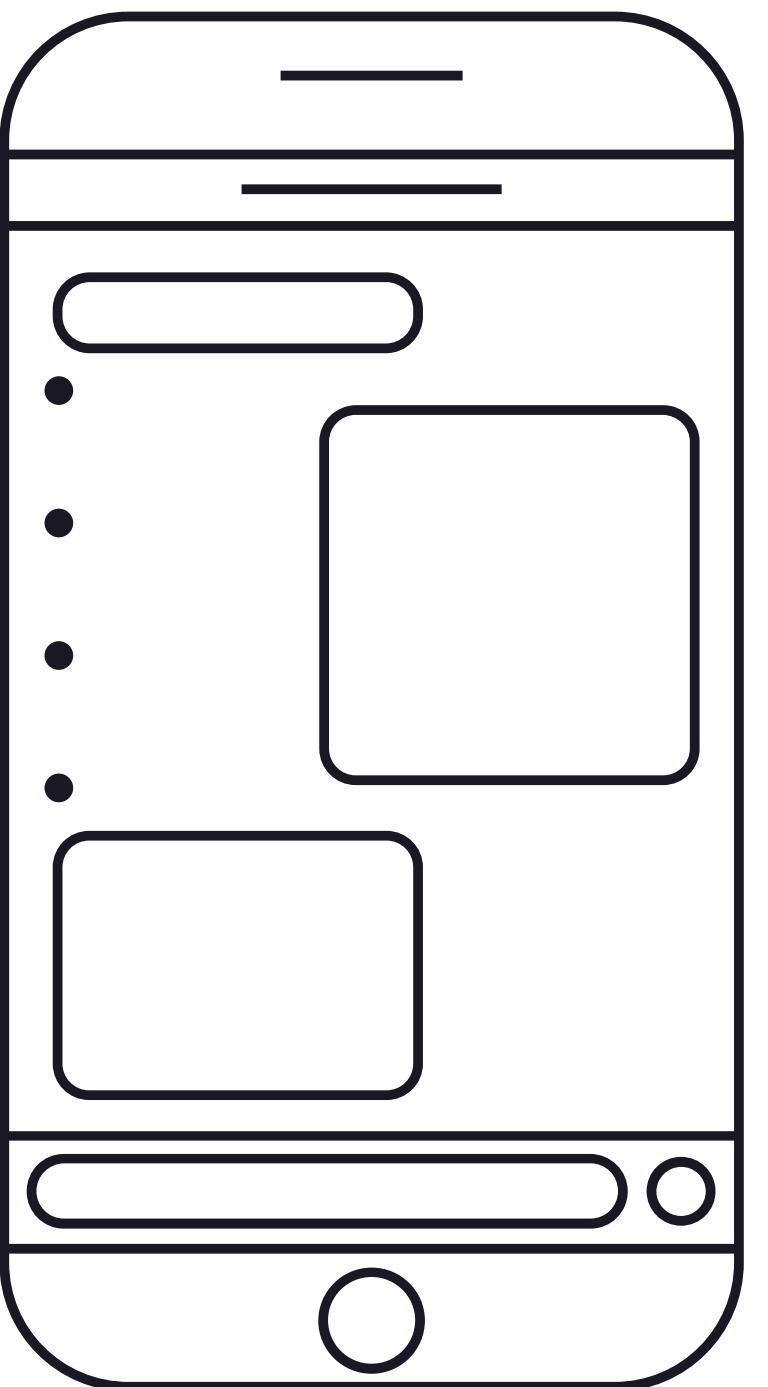
Enqueue(8)



# ANSWER

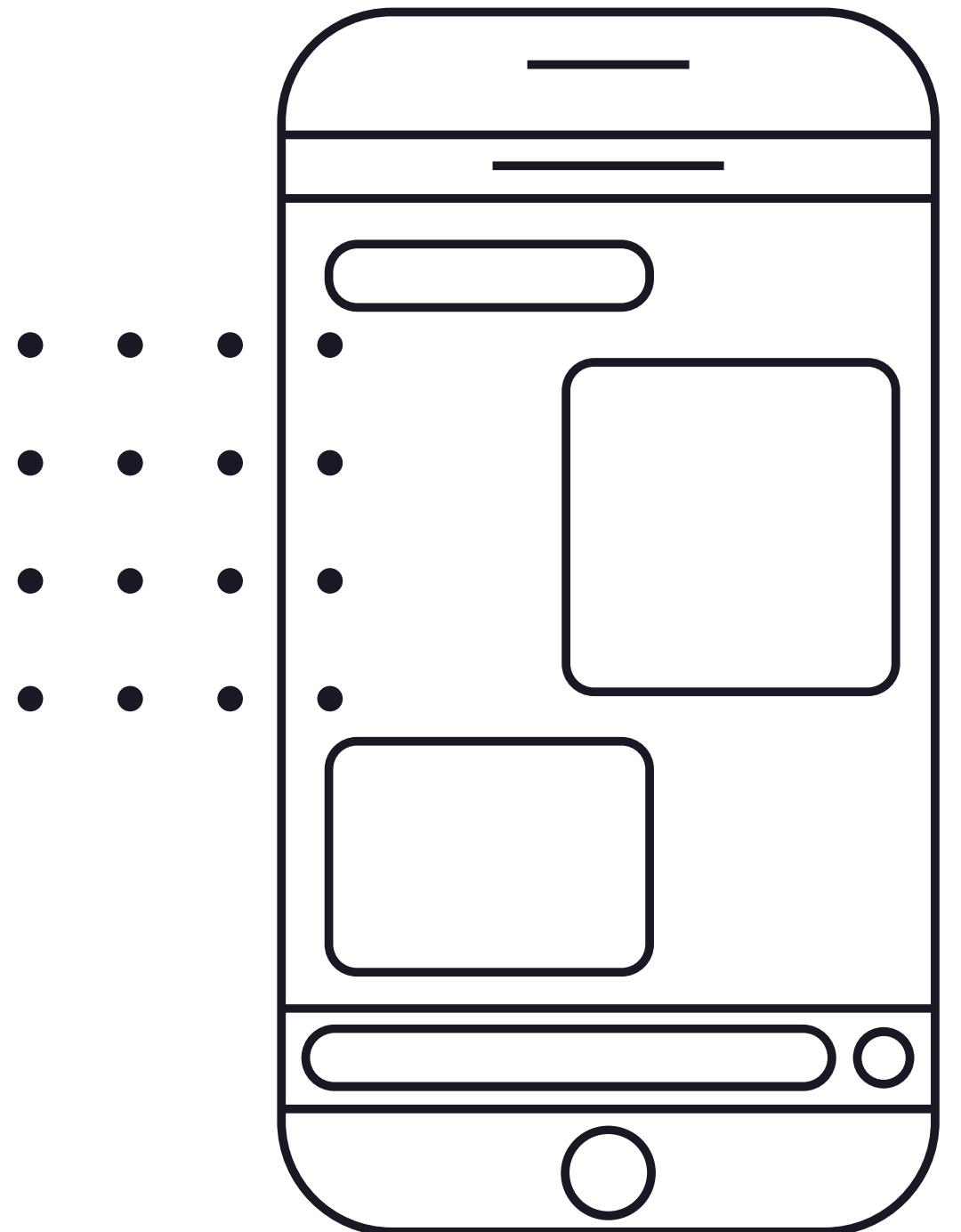
		2	7	8
0	1	2	3	4

• • •  
• • •  
• • •  
• • •  
• • •



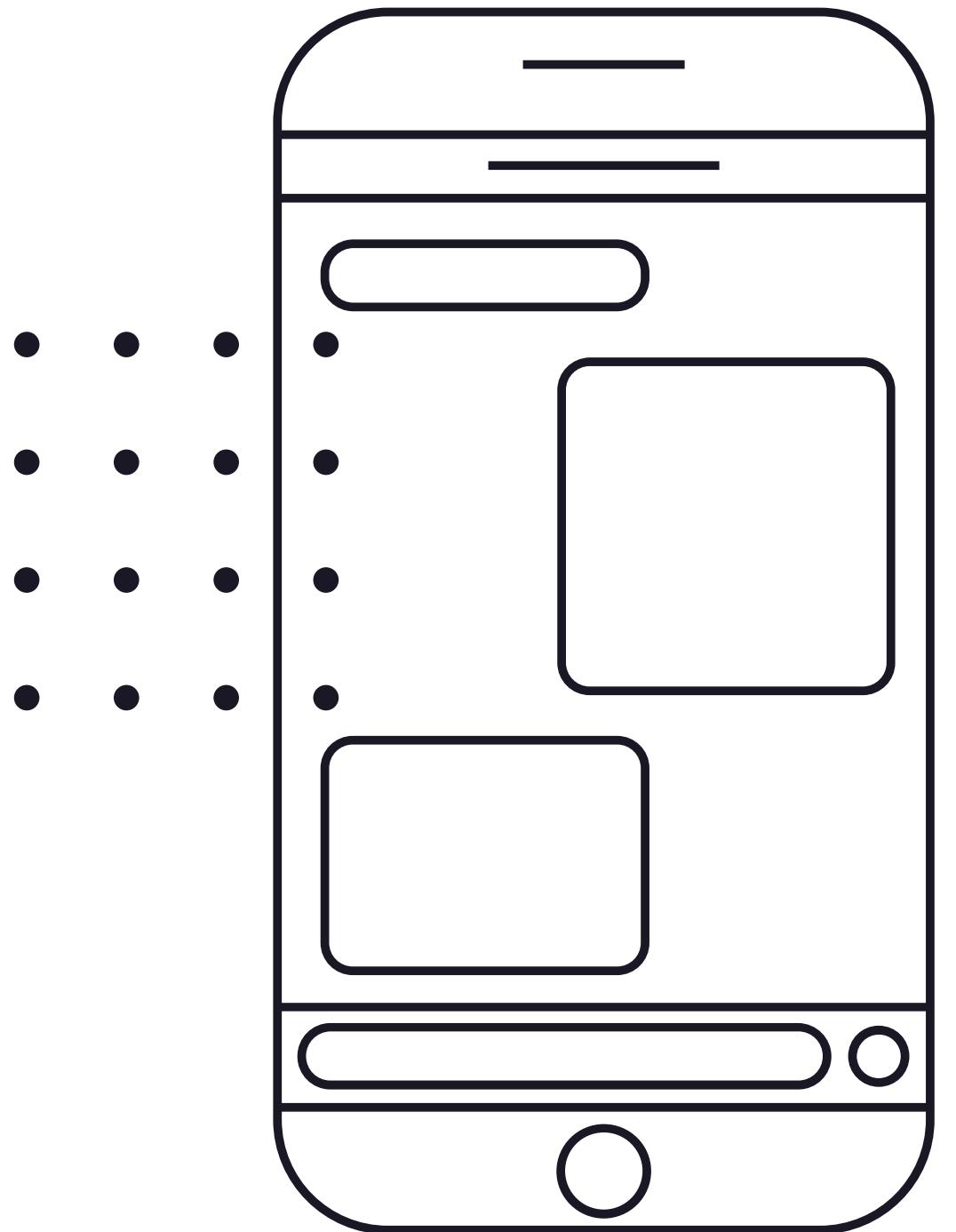
# QUESTION 2

What is the time complexity associated with doing enqueue and dequeue operations in a queue?



# ANSWER

O(1)



# QUESTION 3

08

Consider the following operation where  $k$  is a global parameter. What will be the worst case time complexity of a sequence of  $n$  MultiDequeue() operations if the queue is empty initially?

```
MultiDequeue (Q)
{
    m = k;
    while (Q is not empty and m > 0)
    {
        Dequeue (Q)
        m = m - 1;
    }
}
```

- A.  $O(n)$
- B.  $O(n + k)$
- C.  $O(nk)$
- D.  $O(n^2)$

# ANSWER

08

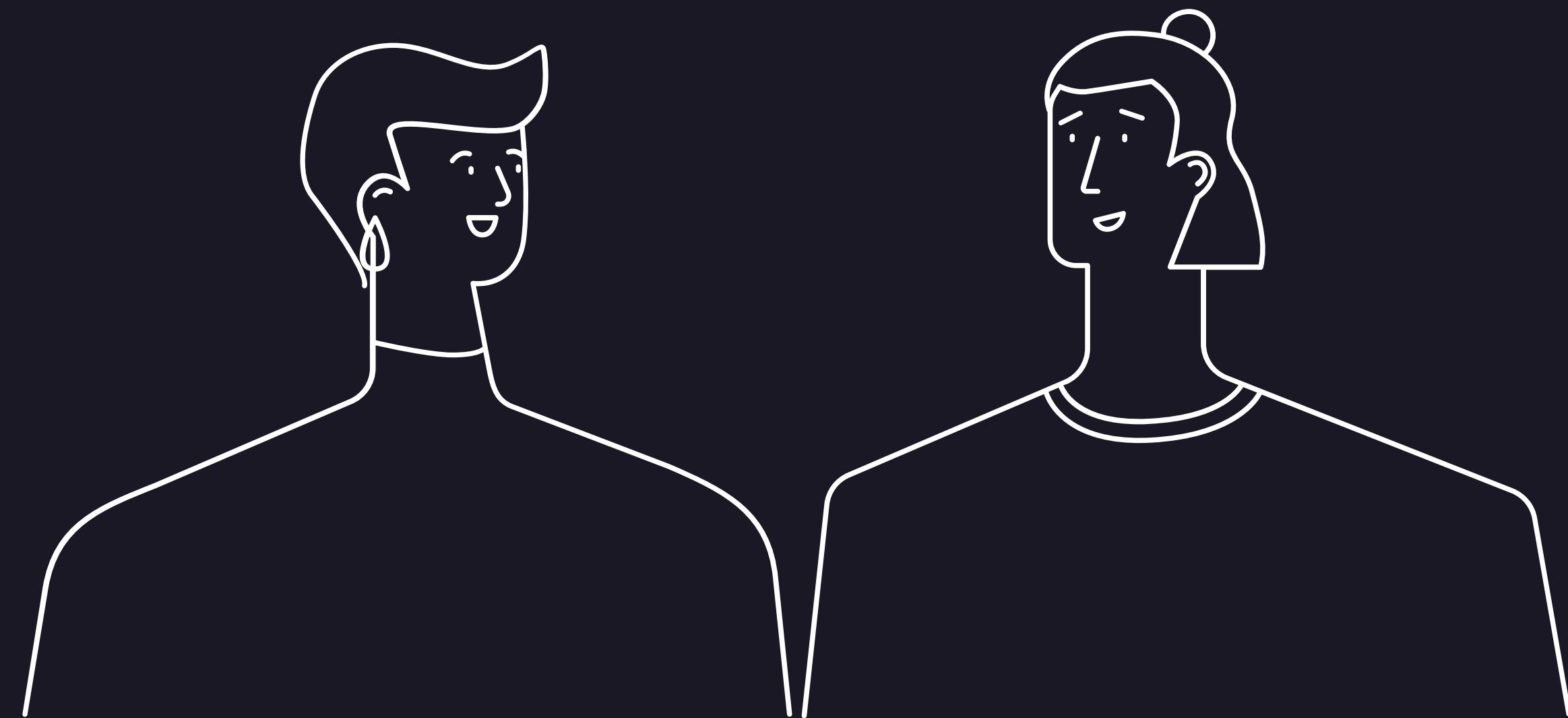
OPTION A,  $O(n)$ . Since the queue is empty initially, the condition of while loop will never become true. So the time complexity will be  $O(n)$ .

```
MultiDequeue (Q)
```

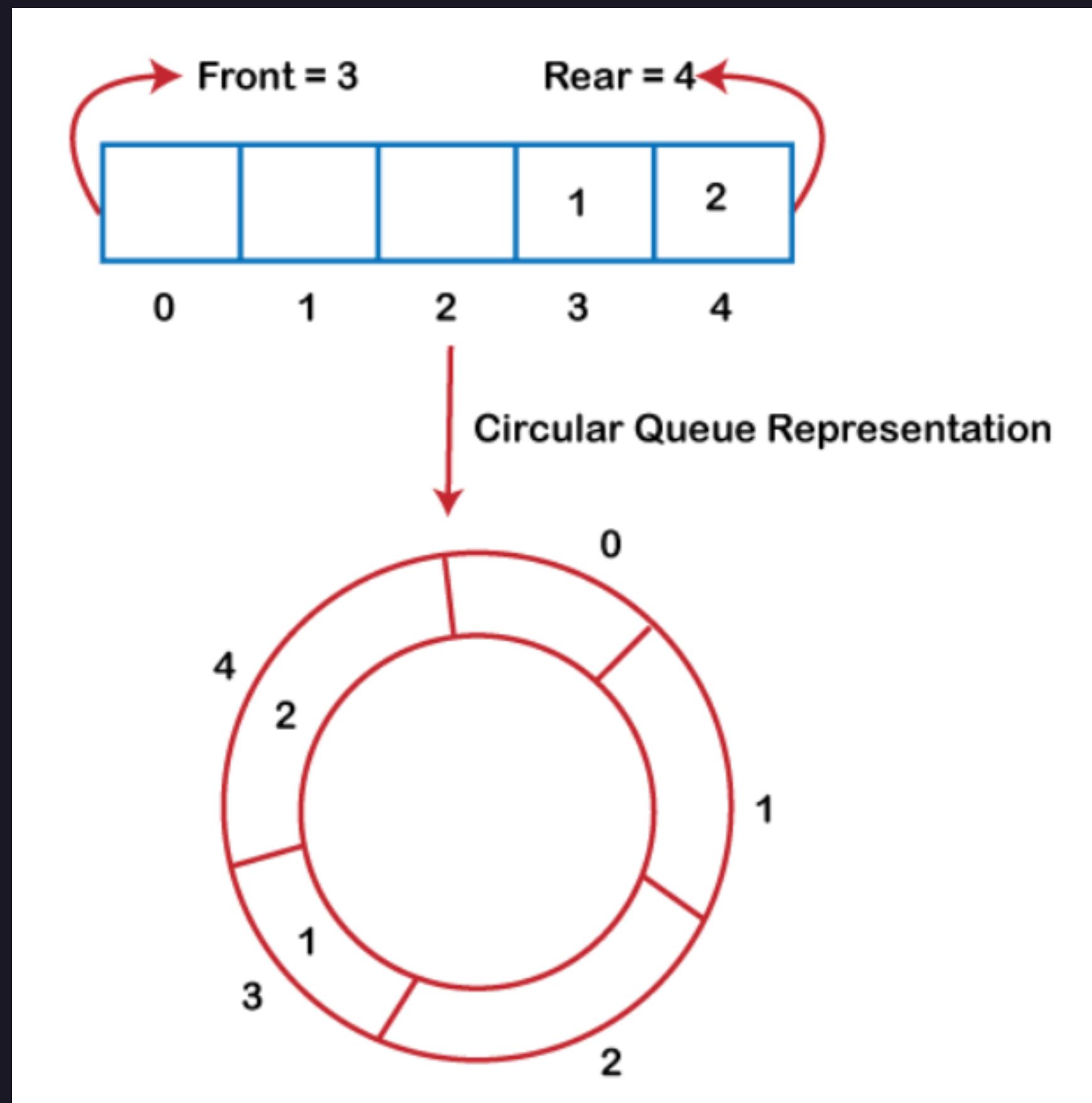
```
{  
    m = k;  
    while (Q is not empty and m > 0)  
    {  
        Dequeue (Q)  
        m = m - 1;  
    }  
}
```

- A.  $O(n)$
- B.  $O(n + k)$
- C.  $O(nk)$
- D.  $O(n^2)$

# PLUS ULTRA: DISCUSSING MORE TYPES OF QUEUES!



# CIRCULAR QUEUE



# Why do we need a circular queue?

# Why do we need a circular queue?

If the rear reaches to the end position of the Queue then there might be possibility that some vacant spaces are left in the beginning which cannot be utilized

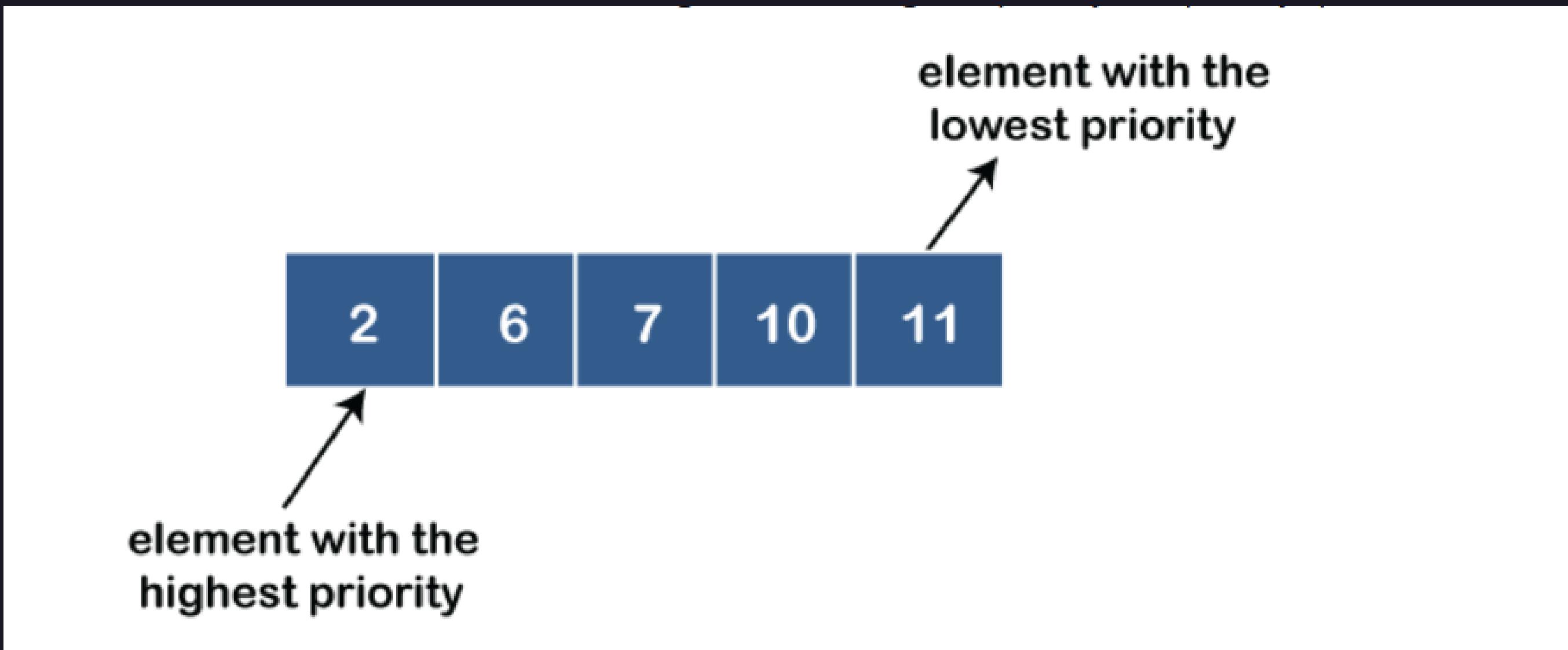
# Priority queue

A Type of queue in which each element has a priority associated with it

# Priority queue

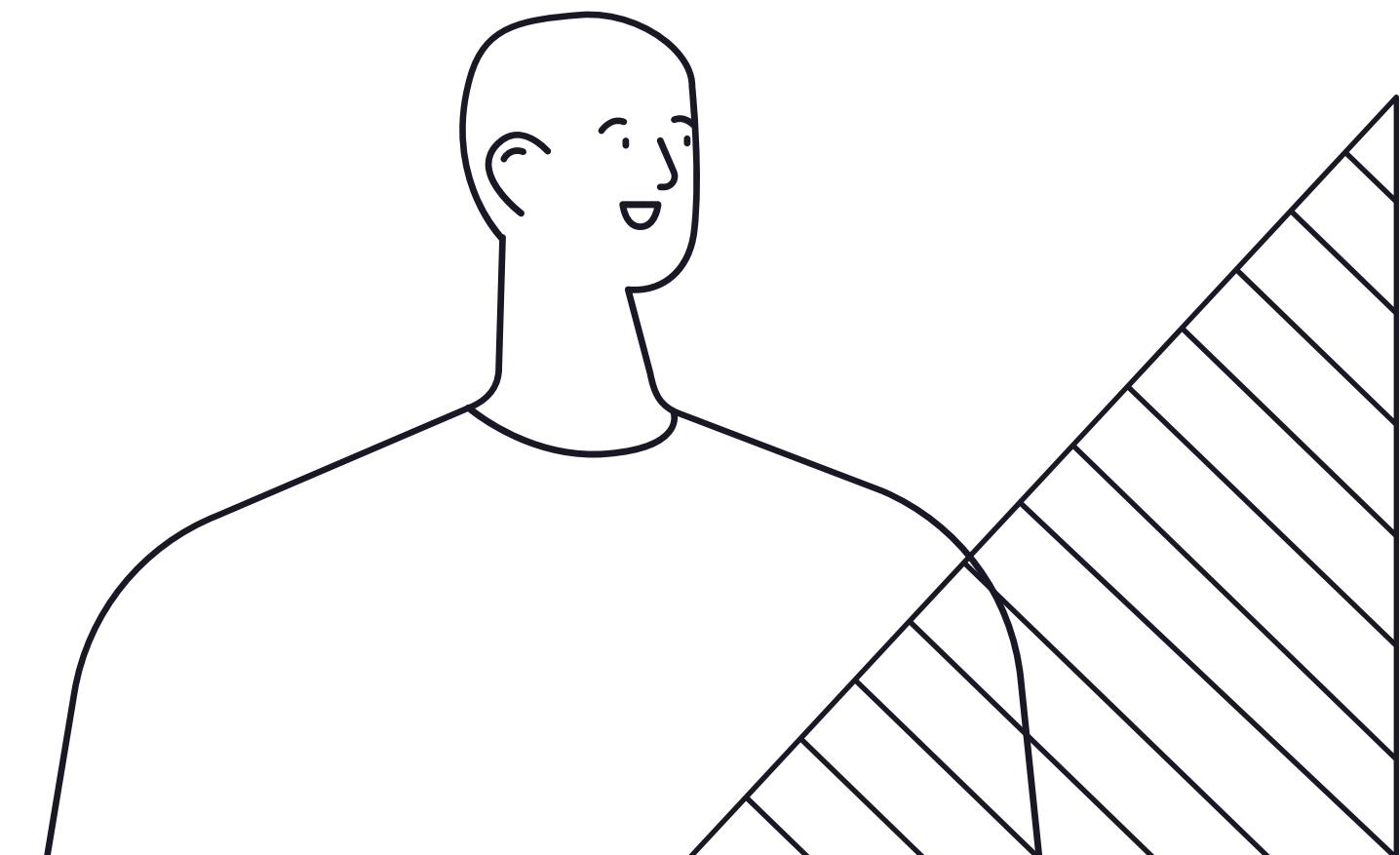
The priority of the elements in a priority queue will determine the order in which elements are removed from the priority queue

# Priority queue



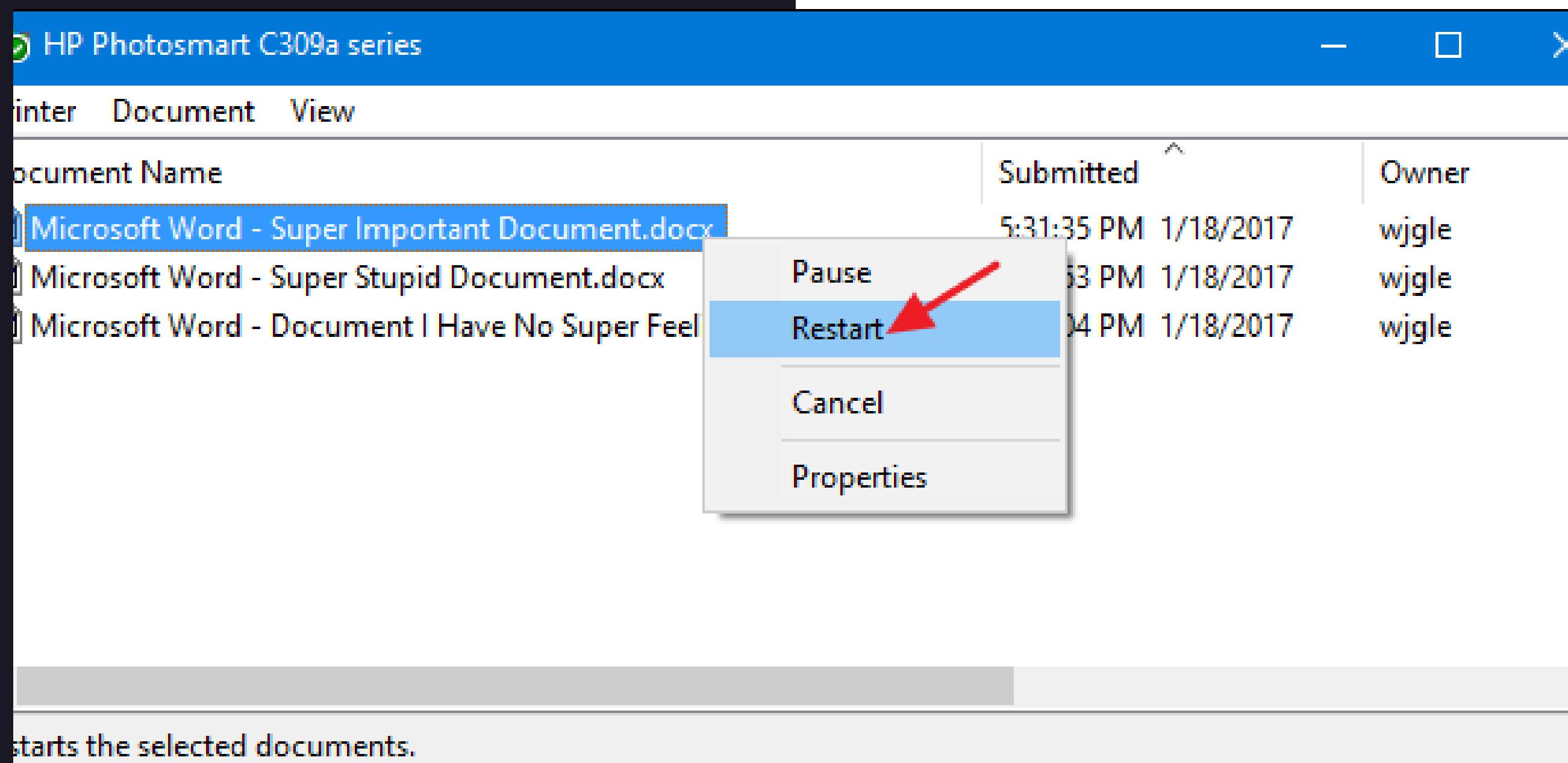
THE THAT'S COOL AND ALL  
PAKSHE WHAT'S IT FOR  
SEGMENT?

# REAL WORLD APPLICATIONS!

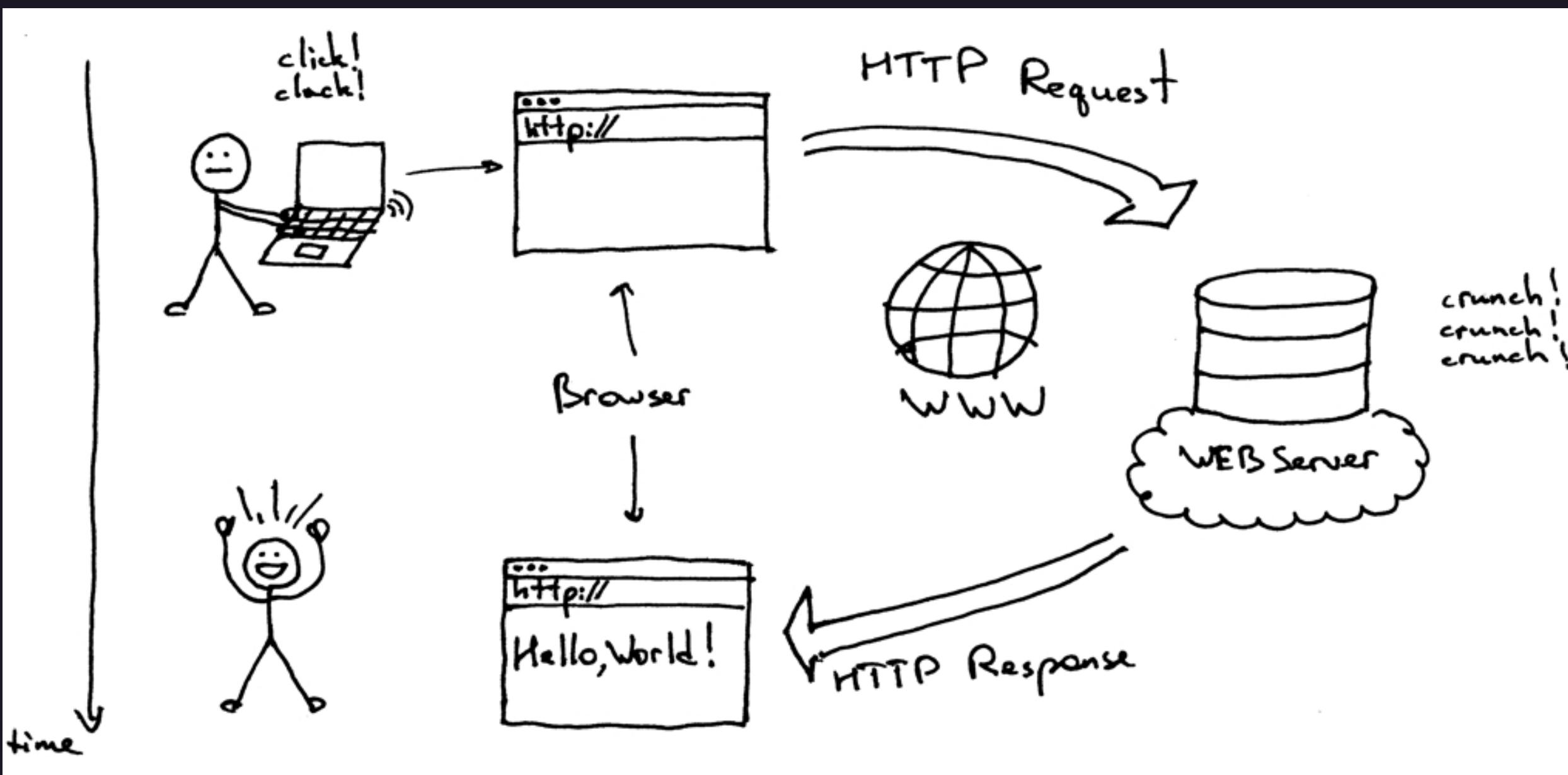


# REAL WORLD APPLICATIONS!

- Queues in mail, router, printers

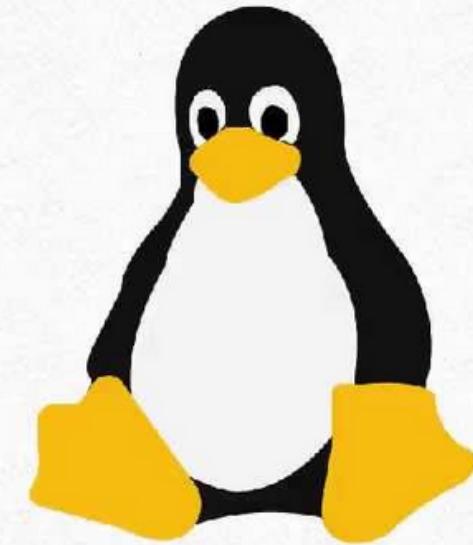
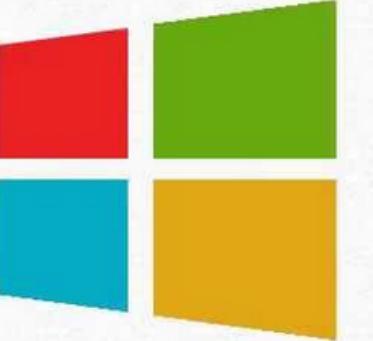


Imagine you have a web-site which serves files to thousands of users. You cannot service all requests, you can only handle say 100 at once. A fair policy would be first-come-first serve: serve 100 at a time in order of arrival. A Queue would definitely be the most appropriate data structure.



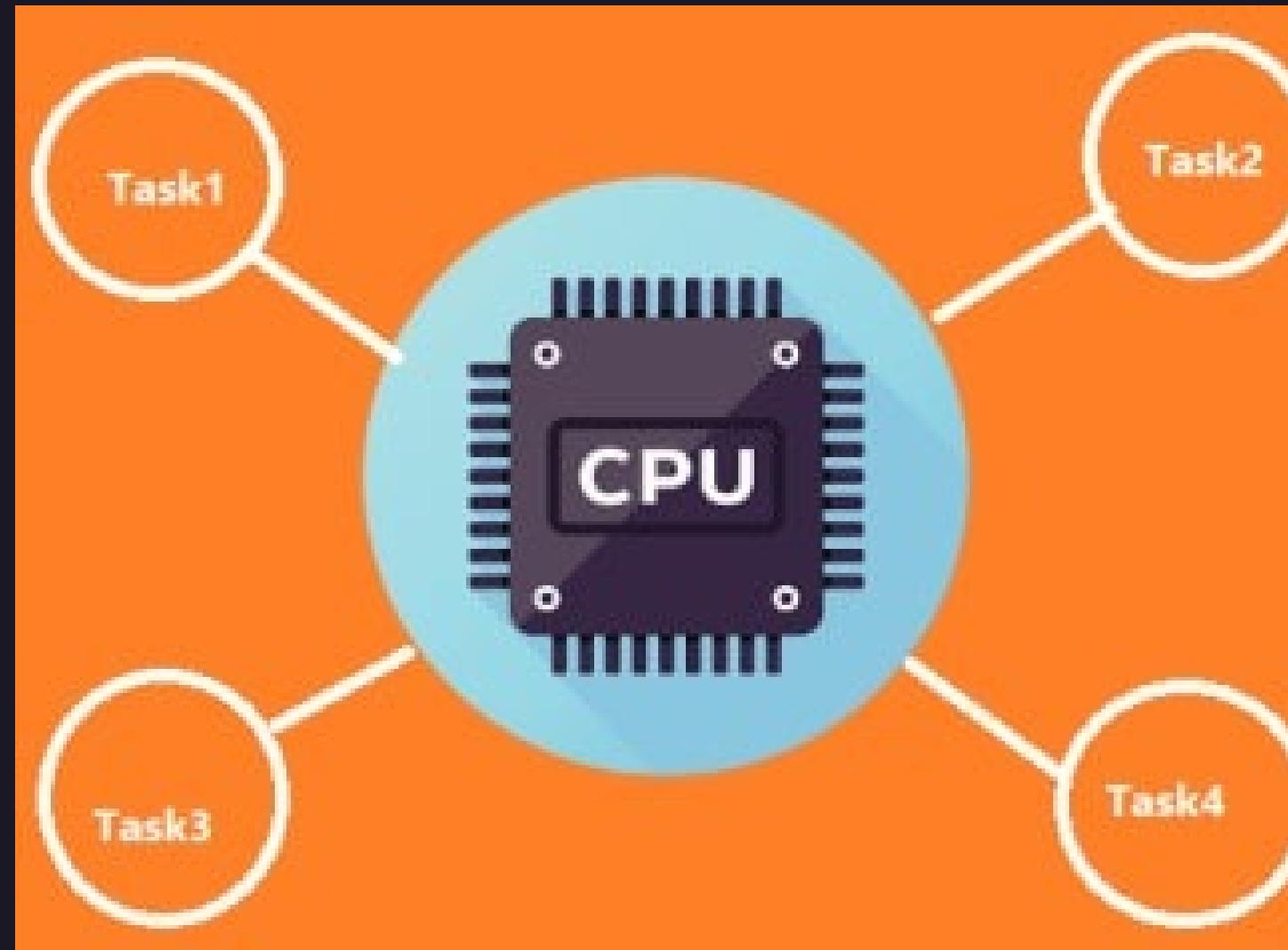
# REAL WORLD APPLICATIONS!

- In Operating systems:
  - a) Semaphores
  - b) FCFS ( first come first serve) scheduling
  - c) CPU scheduling, Disk scheduling



Similarly in a multitasking operating system, the CPU cannot run all jobs at once, so jobs must be batched up and then scheduled according to some policy.

Again, a queue might be a suitable option in this case.





## ADVANCED:TRY IT OUT

Can you implement a Queue using stacks? How many stacks do we need?

# ADVANCED:WHAT NOW?



- Learn DSA online from youtube, geeksforgeeks etc.
- Learn about different data structures like linked lists, trees, graphs

# ADVANCED:WHAT NOW?

11



- Answer easy questions on platforms like
  - Beginner
  - HackerRank
  - Hacker Earth
- Intermediate
- Leetcode
- Codeforces

# Thank you!

I'm Pranav H. Nair

Web Developer | React | RoR | Game developer

Reach out to me @prnvsan on most social media, to connect