

Lab 09 + DSA 02

Stacks and Queues and API Server

seattle-javascript-401n14

Lab 08 Review



Code Challenge 08

Review



Vocab Review!



What is mongoose?



What is mongoose?

Mongoose is a middleman (aka middleware) between our Node.js applications and a MongoDB database. Mongoose helps us impose schema validation upon our database, and helps us properly connect to MongoDB.

What is an endpoint?



What is an endpoint?

An endpoint is a route in the server that clients can request data from. Each endpoint should return a response if there is a properly formatted request.

An endpoint is the end of a communication channel between the client and server.

What is middleware?



What is middleware?

Middleware is code that typically runs between a request and response cycle. This can be very powerful to modify the request, edit how the response is delivered, or more!

What is a Data Model?



What is a Data Model?

A data model is an abstract structure defines a collection of data and standardizes how data relates to one another. For instance, a data model may specify a “car” model and a “driver” model, and define how they’re related. In our applications, our data models will also define our database operations (Create, Read, Update and Delete)

What is a stack?



What is a stack?

A stack is a data structure where insertion and deletion of items takes place at one end called the **top of the stack**. The basic concept can be illustrated by thinking of your data set as a stack of plates or books where you can only take the top item off the stack in order to remove things from it. This structure is used all throughout programming.

What is a queue?



What is a queue?

A queue is an abstract data structure, somewhat similar to stacks. However unlike stacks, a queue is open at both its ends. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue).

What is big-o?



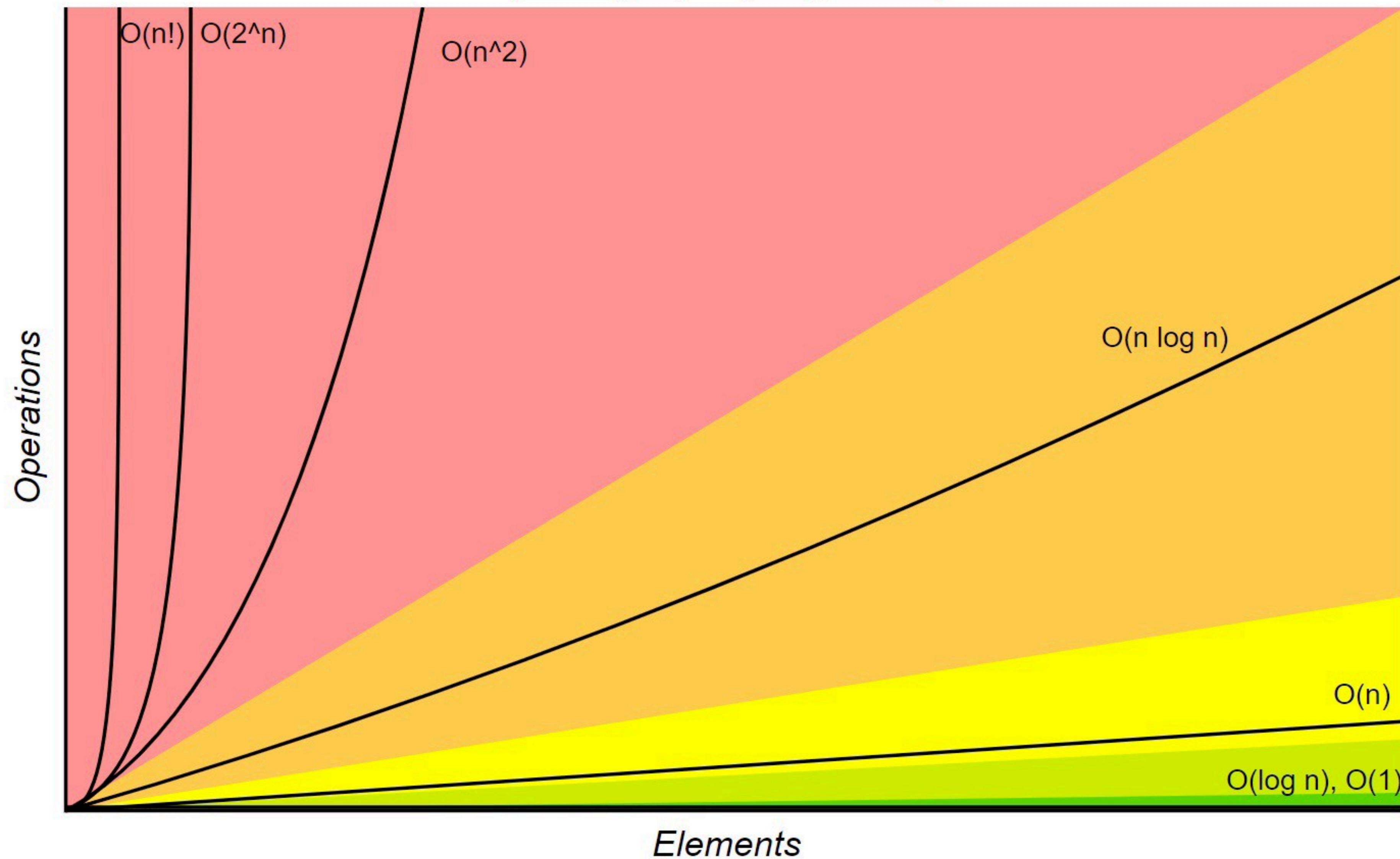
What is big-o?

We use big-o notation to describe the **worst-case behavior** of a function.

Big-o describes how the time and space of a function grows in relation to how the size of the input (n) grows. Standard growth rates are constant (1), logarithmic ($\log n$ or $n \log n$), 1:1 (n), squared (n^2), exponential (2^n) and factorial ($n!$)

Big-O Complexity Chart

Horrible Bad Fair Good Excellent



The Stack Data Structure

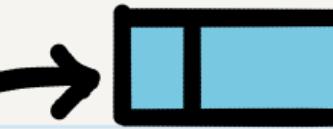
- We can see stacks everywhere in programming
- Call stack is the most common
 - Functions you want to execute are pushed on the stack, and only popped when they're done and at the top
- “Stack Overflow” is when you keep pushing to the stack
- Push and Pop operations are $O(1)$, but what about search?



Push



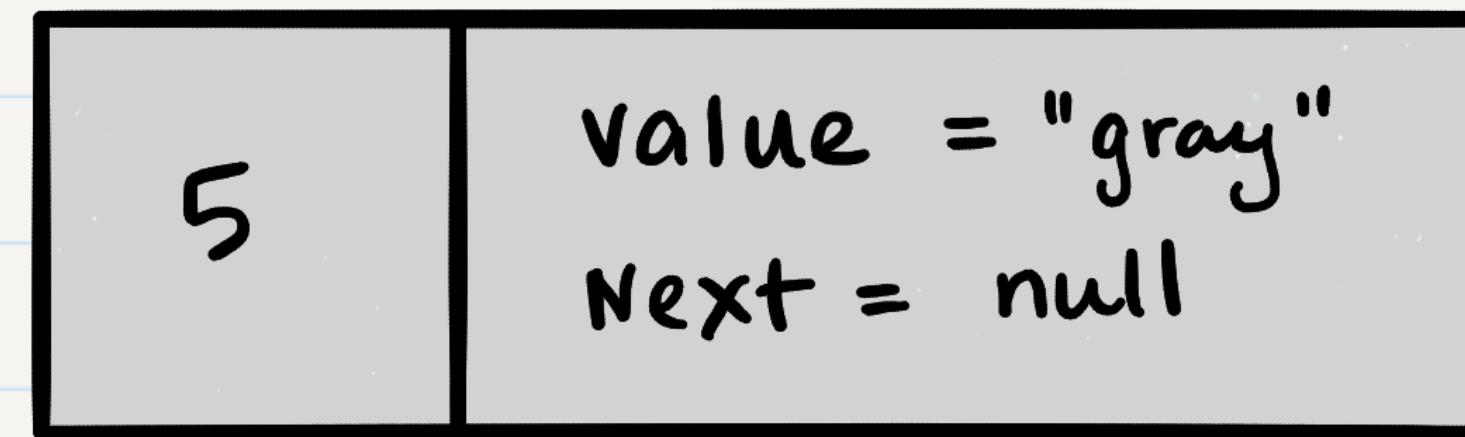
Pop



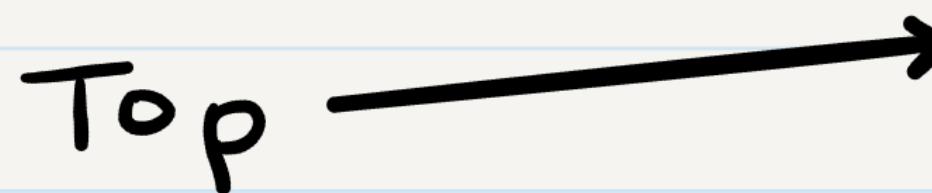
Top



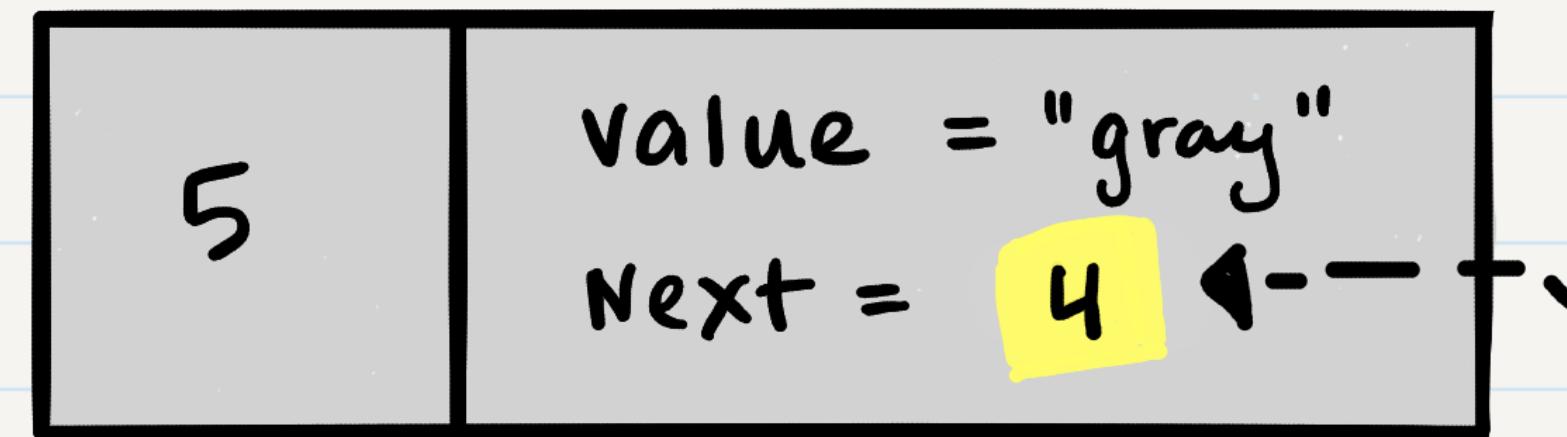
Push :



Top

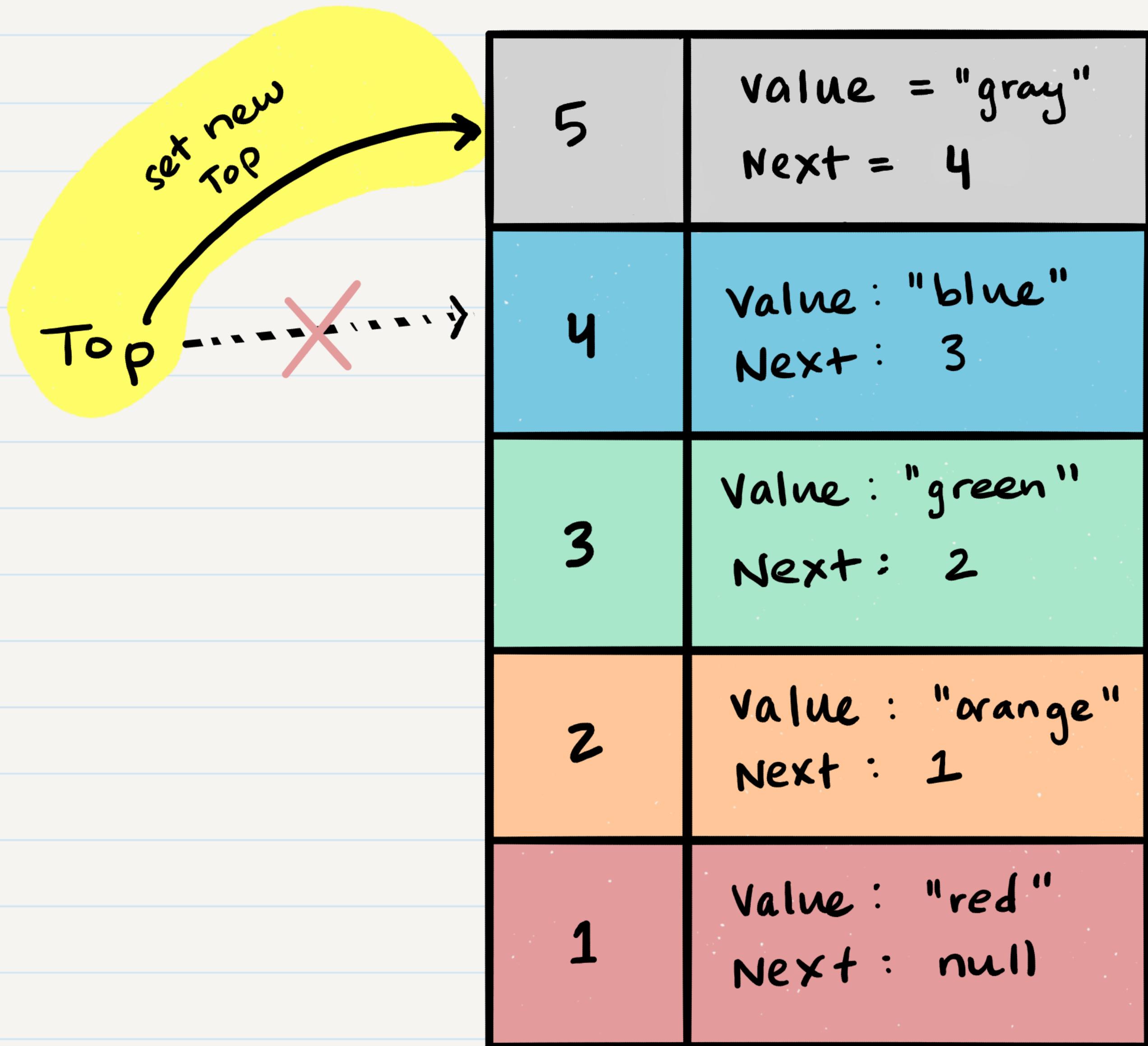


Push :



Top

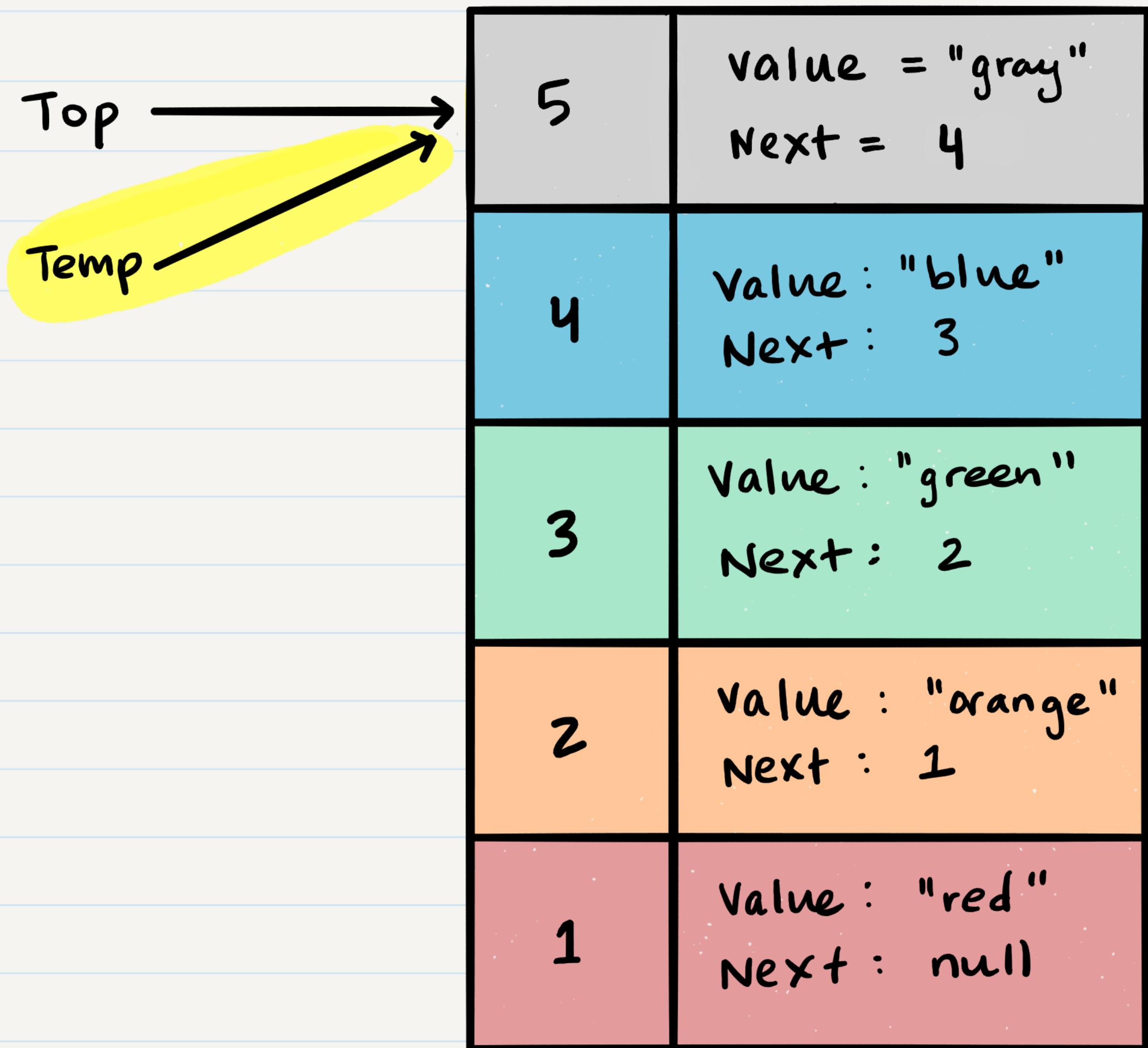


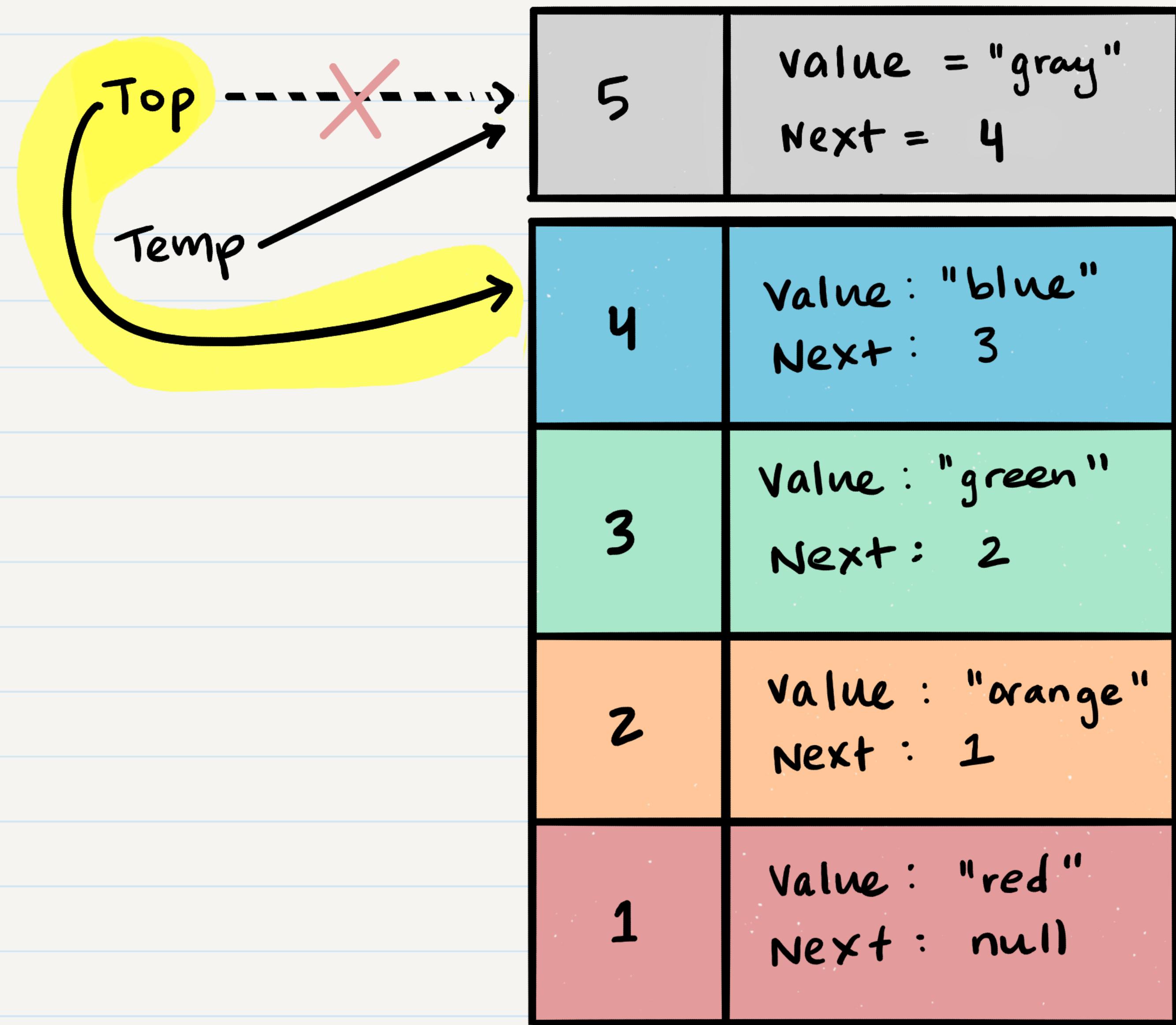


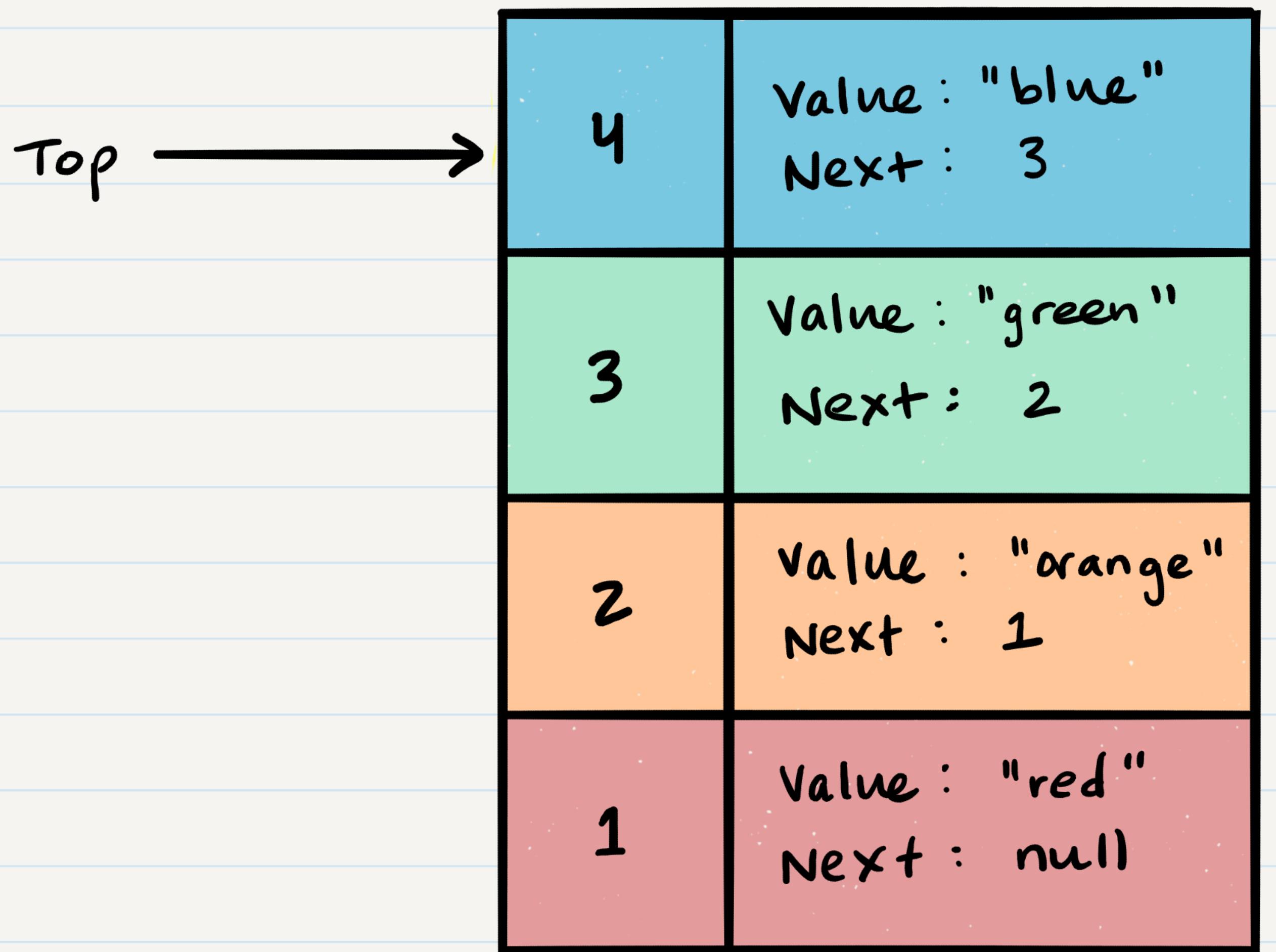
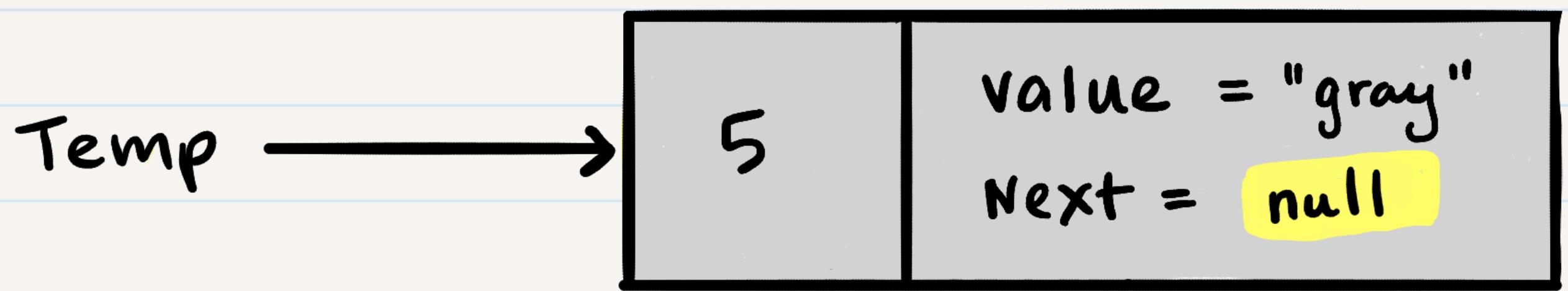
Top



5	value = "gray" Next = 4
4	value : "blue" Next : 3
3	Value : "green" Next: 2
2	value : "orange" Next : 1
1	Value : "red" Next : null







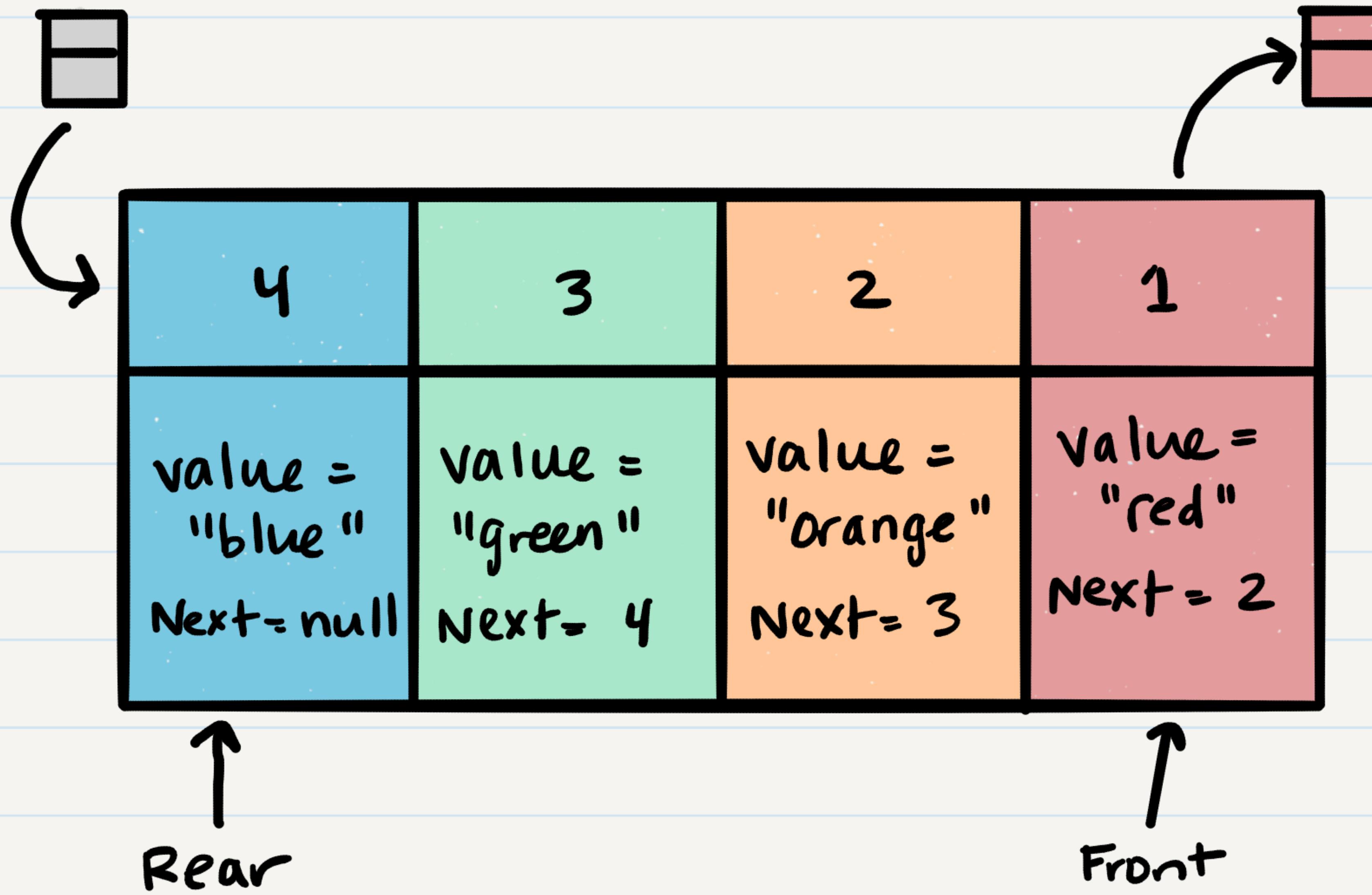
Stacks are
Last In First Out
Or
First In Last Out

The Queue Data Structure

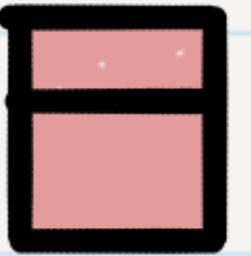
- Queues have a bit more to manage; they need to know their front and their back
 - Pop the front, Push to the back
- Again, since we're just storing the front and back, enqueue and dequeue are $O(1)$



Enqueue

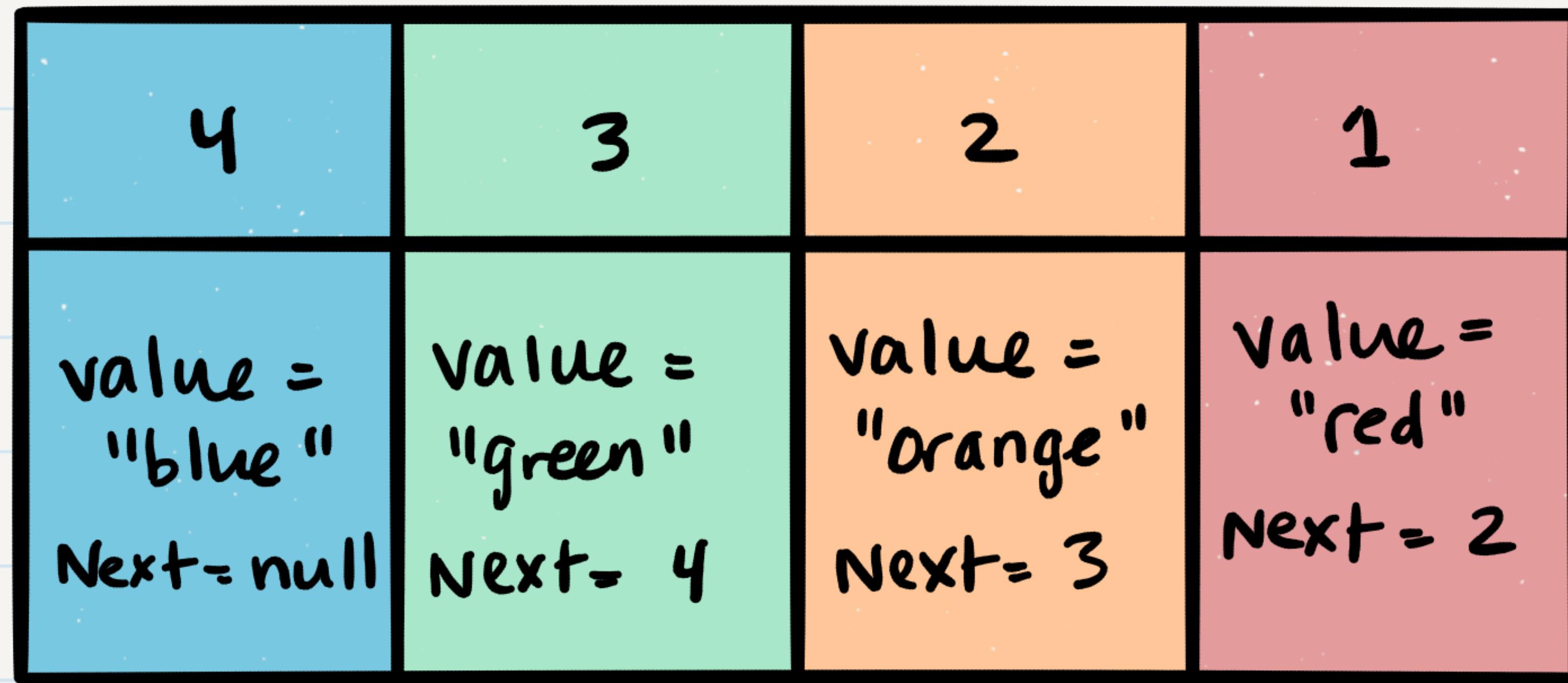


Dequeue



Enqueue
↓

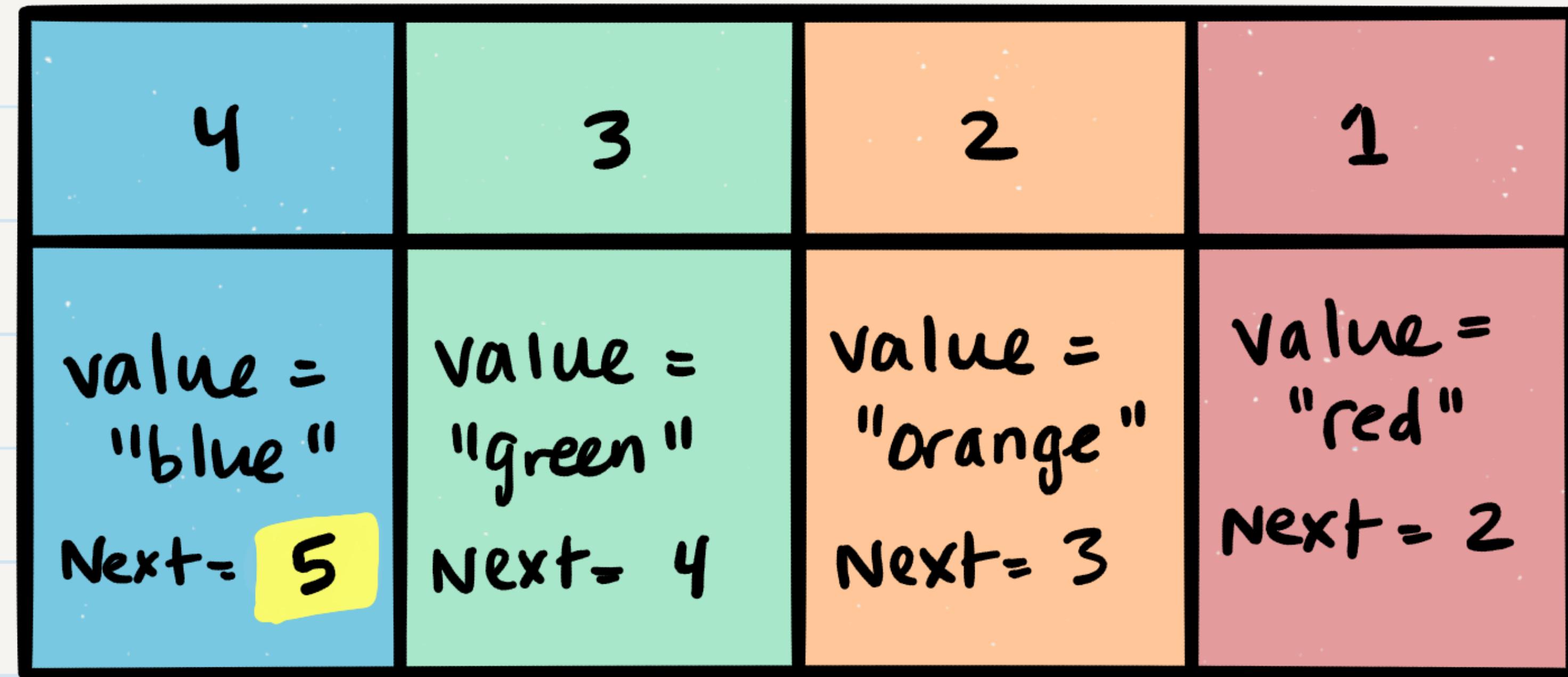
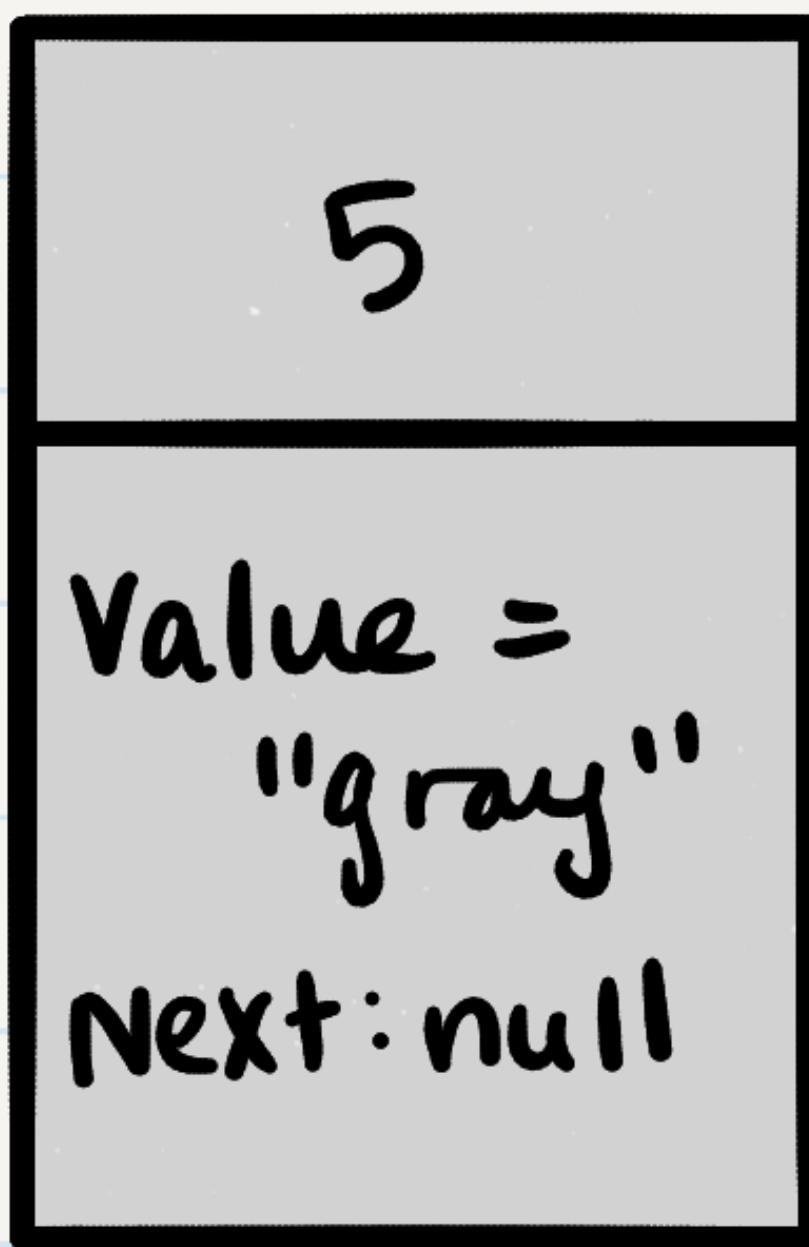
5
Value = "gray"
Next: null



↑
Rear

↑
Front

Enqueue



↑
Rear

↑
Front

Enqueue
↓

5	4	3	2	1
Value = "gray" Next: null	value = "blue" Next = 5	value = "green" Next = 4	value = "orange" Next = 3	value = "red" Next = 2

Rear

Front

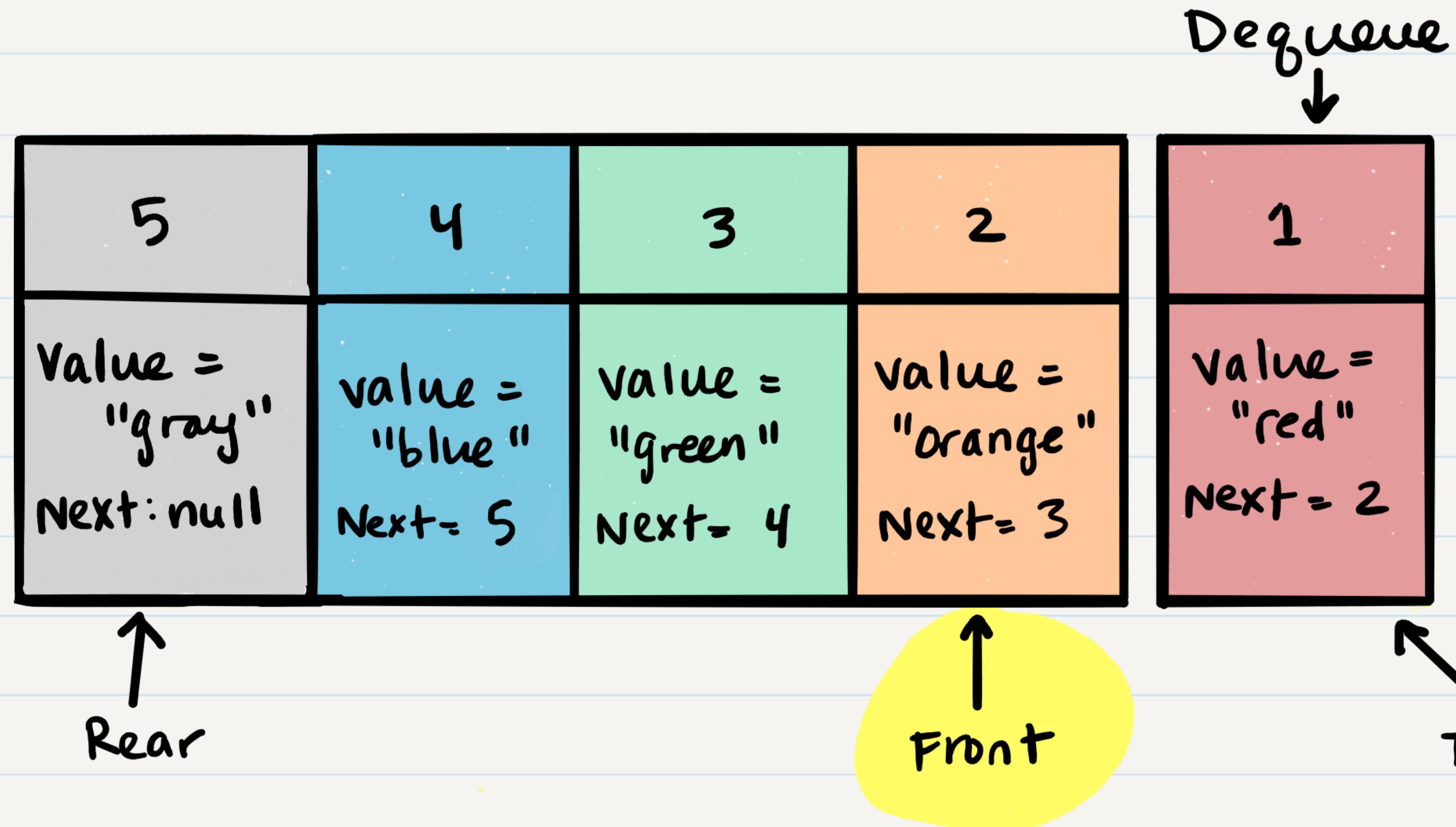
Dequeue
↓

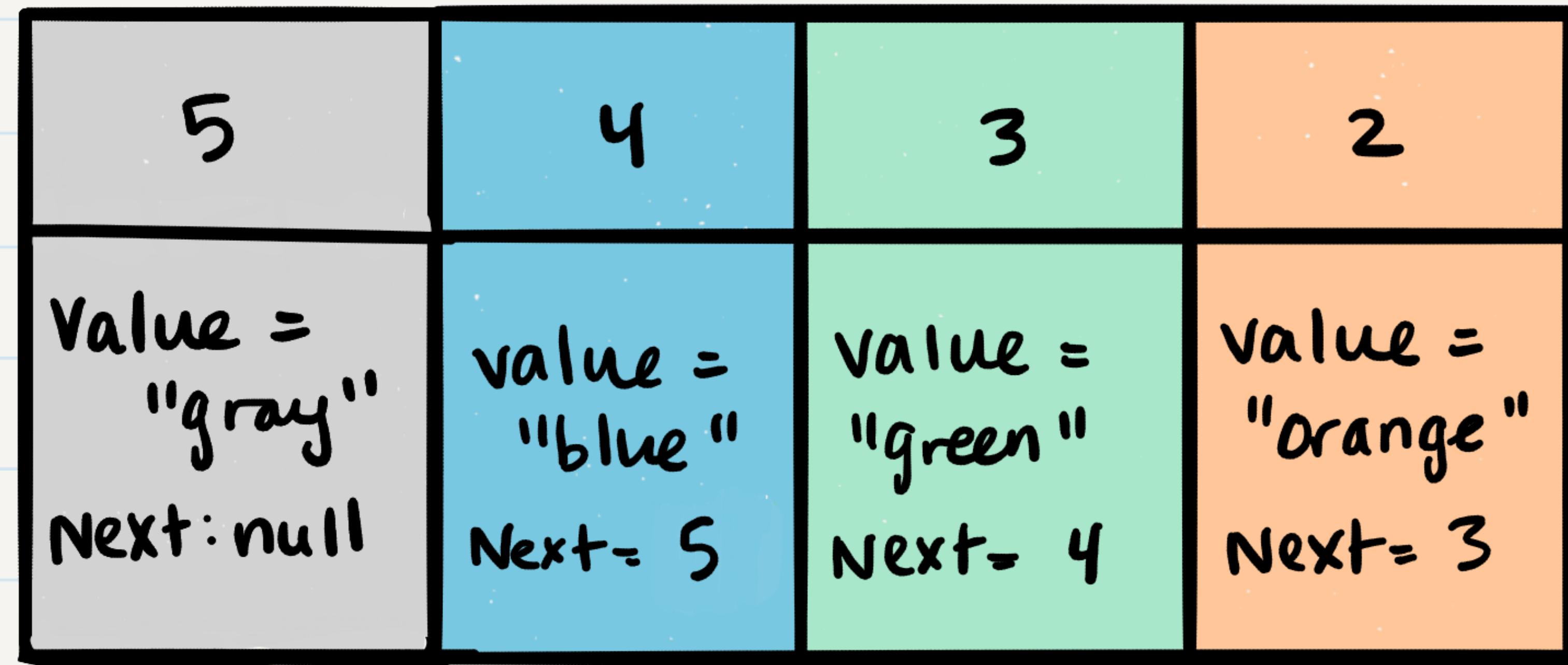
5	4	3	2	1
Value = "gray" Next: null	value = "blue" Next= 5	value = "green" Next= 4	value = "orange" Next= 3	value = "red" Next = 2

↑
Rear

↑
Front

Temp





↑

Rear

↑

Front



↑
Temp

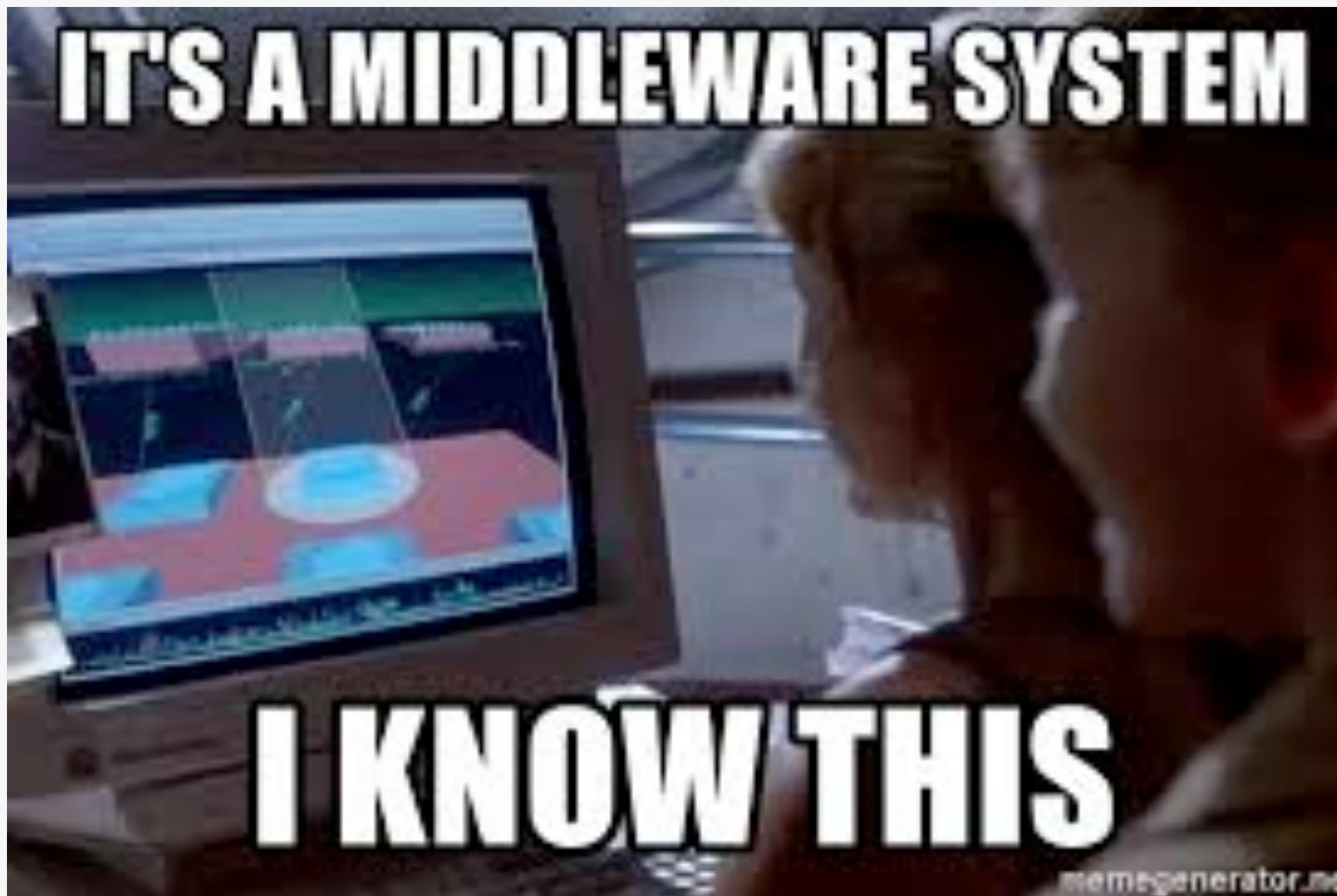
Queues are
Last In Last Out
Or
First In First Out

Break!



More about Mongoose

- We have been using mongoose for schema validation, but it can do so much more!
- You can define virtual properties that don't exist in the schema, but can still be returned for each record
- Mongoose is a **client** for the database **server**
 - Because this is a client -> server relationship, we can add **middleware**!



Virtual Properties

- **Virtual properties** are fake key-value pairs, where the value is **generated by a function** instead of found within an object
 - Because a virtual property is not a real key-value pair, we have to define a **getter** and a **setter**

```
schema.virtual(keyname).get(myFunc)
```

```
schema.virtual(keyname).set(myFunc)
```



Virtual Properties

```
32` peopleSchema.virtual('fullName').get(function() {  
33    console.log('GETTING A FAKE THING');  
34    return this.firstName + ' ' + this.lastName;  
35});  
36  
37` peopleSchema.virtual('fullName').set(function(name) {  
38    let str = name.split(' ');  
39  
40    this.firstName = str[0];  
41    this.lastName = str[1];  
42});
```



Mongoose Middleware

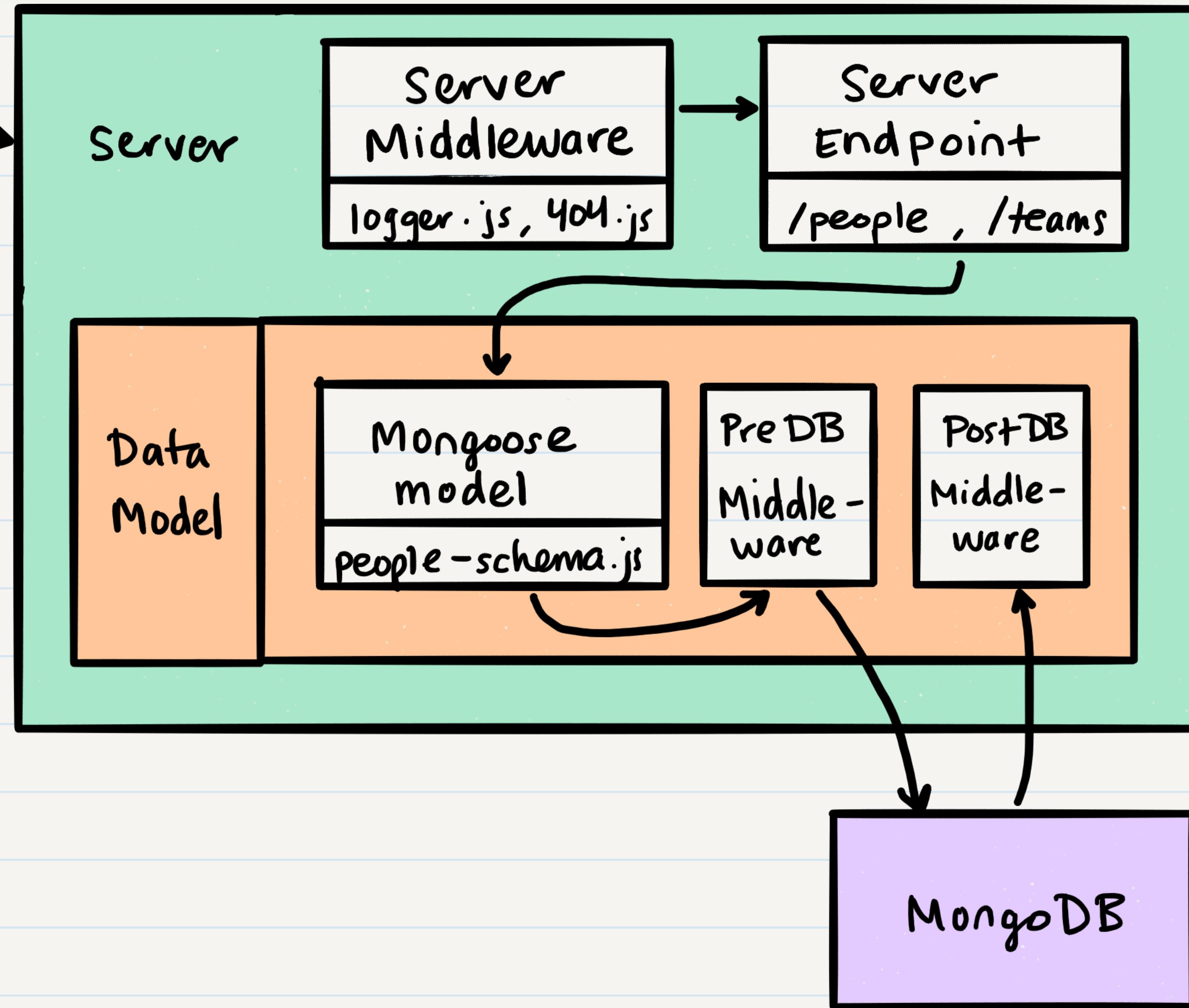
- Mongoose exposes two ways to run middleware: using **pre** and **post** hooks
- A pre-hook middleware will run BEFORE a corresponding database command has been called

```
model.pre('<MongoDB Command>', myFunc);
```

- A post-hook middleware will run AFTER a corresponding database command has been called

```
model.post('<MongoDB Command>', myFunc);
```





Mongoose Middleware

```
const updated = function(next) {
  let now = Date.now();
  this.lastUpdated = now;
  next();
};

peopleSchema.pre('save', updated);
peopleSchema.pre('updateOne', updated);
```

```
const saveLogger = function(next) {
  console.log('Saved Record');
  next();
};

peopleSchema.post('save', saveLogger);
```



Schedule Instructor 1:1

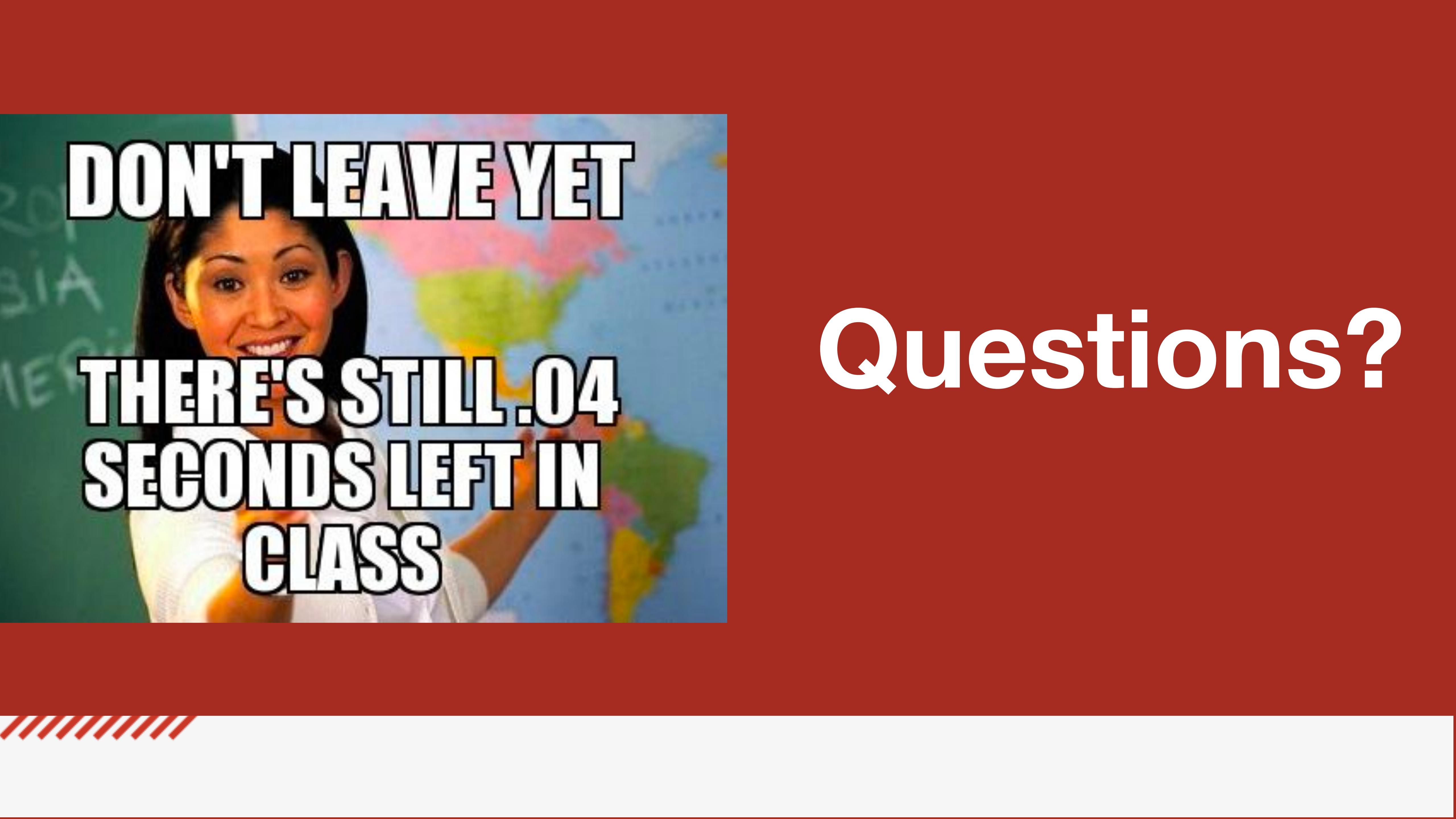
- We're 25% done with the course! 🎉
- You have an assignment due on Sunday, Nov 03 at Midnight:
[Instructor 1:1 Sync](#)
- Let's schedule some time to chat! You can book something here:
[soniakandah.youcanbook.me](#)



What's Next:

- Due by Midnight Tonight:
 - **Partner Power Hour Report #2**
 - **Learning Journal 09**
- Due by Midnight Sunday:
 - **Feedback Week 5**
 - **Career Coaching Assignments**
- Due by Midnight Monday:
 - **Code Challenge 09**
- Next Class on **MONDAY** :
 - **Class 10 - Authentication**



A photograph of a young woman with dark hair, smiling broadly. She is wearing a white long-sleeved shirt. Behind her is a large world map on a wall, showing various continents in different colors. To the left of the map, a green chalkboard is visible with some faint, illegible writing.

DON'T LEAVE YET

THERE'S STILL .04

SECONDS LEFT IN

CLASS

Questions?