DATA STRUCTURES

“It's easy to make mistakes that only come out much later, after you've already implemented a lot of code. You'll realize Oh I should have used a different type of data structure. Start over from scratch.”

Guildo van Rossum

(author of Python)

Overview

1. What is a Data Structure?
2. Why are Data Structures important?
3. Data Structure examples
    a. Linked Lists
4. What are Linked Lists?
5. Live Coding of a Linked List

What are data structures?

“In computer science, a data structure is a data organization, management, and storage format that enables efficient access and modification. More precisely, **a data structure is a collection of data values, the relationships among them, and the functions or operations that can be applied to the data.**”

- Wikipedia, “Data Structure”

“An organization of information, usually in memory, for better algorithm efficiency, such as queue, stack, linked list, heap, dictionary, and tree, or conceptual unity, such as the name and address of a person. It may include redundant information, such as length of the list or number of nodes in a subtree.”

- *Paul E. Black, "data structure", in Dictionary of Algorithms and Data Structures*

The way you organize your data

Why are data structures important?

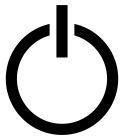
- Using the right data structure will help your software run more efficient.
- … you manipulate or change data more efficient.
- … you scale your application.
- … you become a better engineer.

Makes your software more efficient

Data Structure examples

Data Types

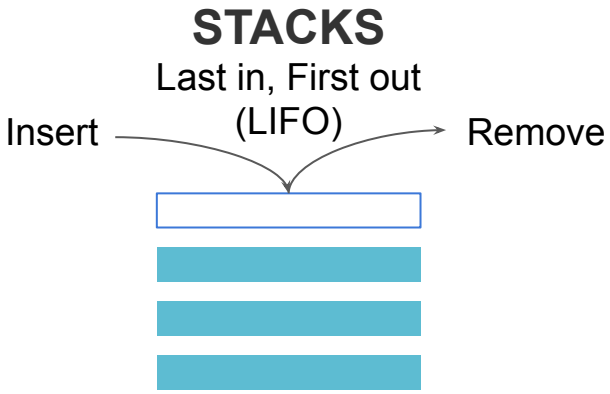
1. Strings
2. Numbers
3. Arrays/Lists
4. Objects/Dictionaries
5. Boolean (true | false)
6. null/None (has no value)
7. undefined (no value assigned)

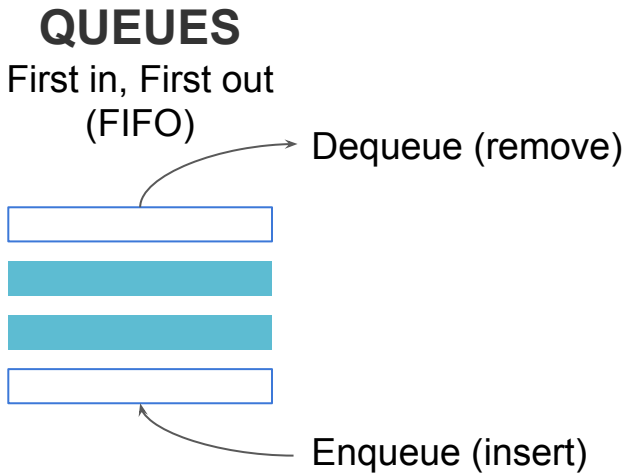


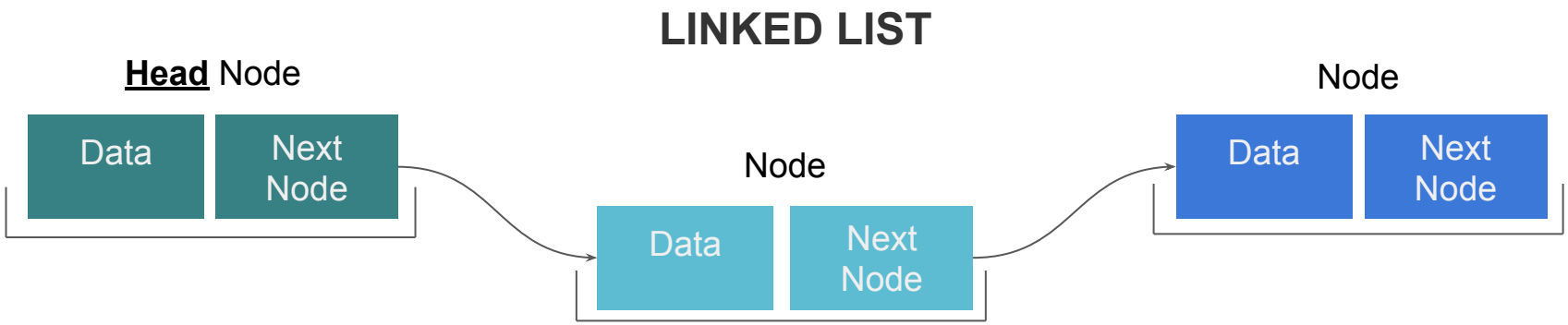
Data Structures

1. Array/List
2. Object/Dictionary
3. Stacks
4. Queues
5. **Linked Lists** (& Doubly Linked Lists)
6. Binary Search Trees
7. Trees
8. Hash Tables
9. Tries
10. Min/Max Heaps
11. and more...

Stacks, Queues, & Linked Lists







Why do Link Lists Exist?

The Problem

```
1   let arrOne = [4,7,2,8,6,9];
2
```

Visual representation of RAM (Random Access Memory)

Why do Link Lists Exist?

The Problem

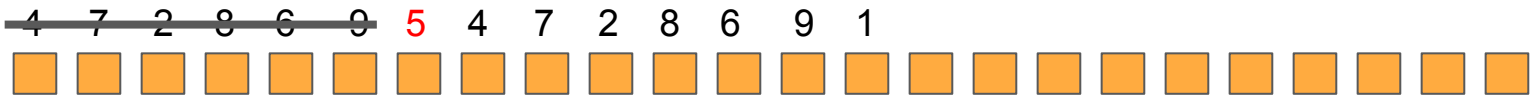
```
1   let arrOne = [4,7,2,8,6,9];
2
3   let intOne = 5;
4
5   arrOne.push(1);
6
7   console.log(arrOne) // [4,7,2,8,6,9,1]
```

Visual representation of RAM (Random Access Memory)

Why do Link Lists Exist?

The Problem

```
1   let arrOne = [4,7,2,8,6,9];
2
3   let intOne = 5;
4
5   arrOne.push(1);
6
7   console.log(arrOne) // [4,7,2,8,6,9,1]
```

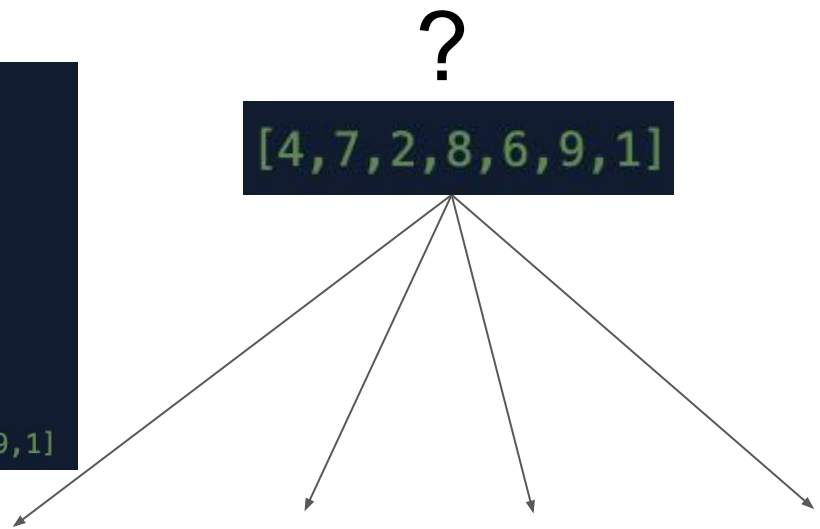


Visual representation of RAM (Random Access Memory)

Why do Link Lists Exist?

The Problem

```
1   let arrOne = [4,7,2,8,6,9];
2
3   let intOne = 5;
4   let intTwo = 3;
5   let intThree = 8;
6   let intFour = 2;
7
8   arrOne.push(1);
9
10  console.log(arrOne) // [4,7,2,8,6,9,1]
```



Visual representation of RAM (Random Access Memory)

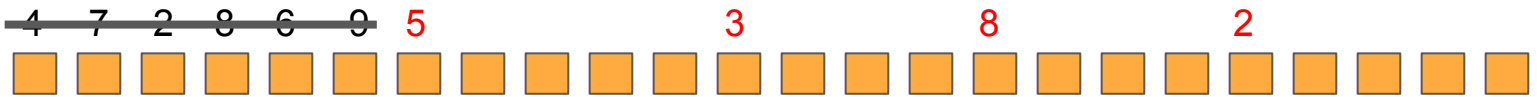
Why do Link Lists Exist?

The Problem

```
1   let arrOne = [4,7,2,8,6,9];
2
3   let intOne = 5;
4   let intTwo = 3;
5   let intThree = 8;
6   let intFour = 2;
7
8   arrOne.push(1);
9
10  console.log(arrOne) // [4,7,2,8,6,9,1]
```

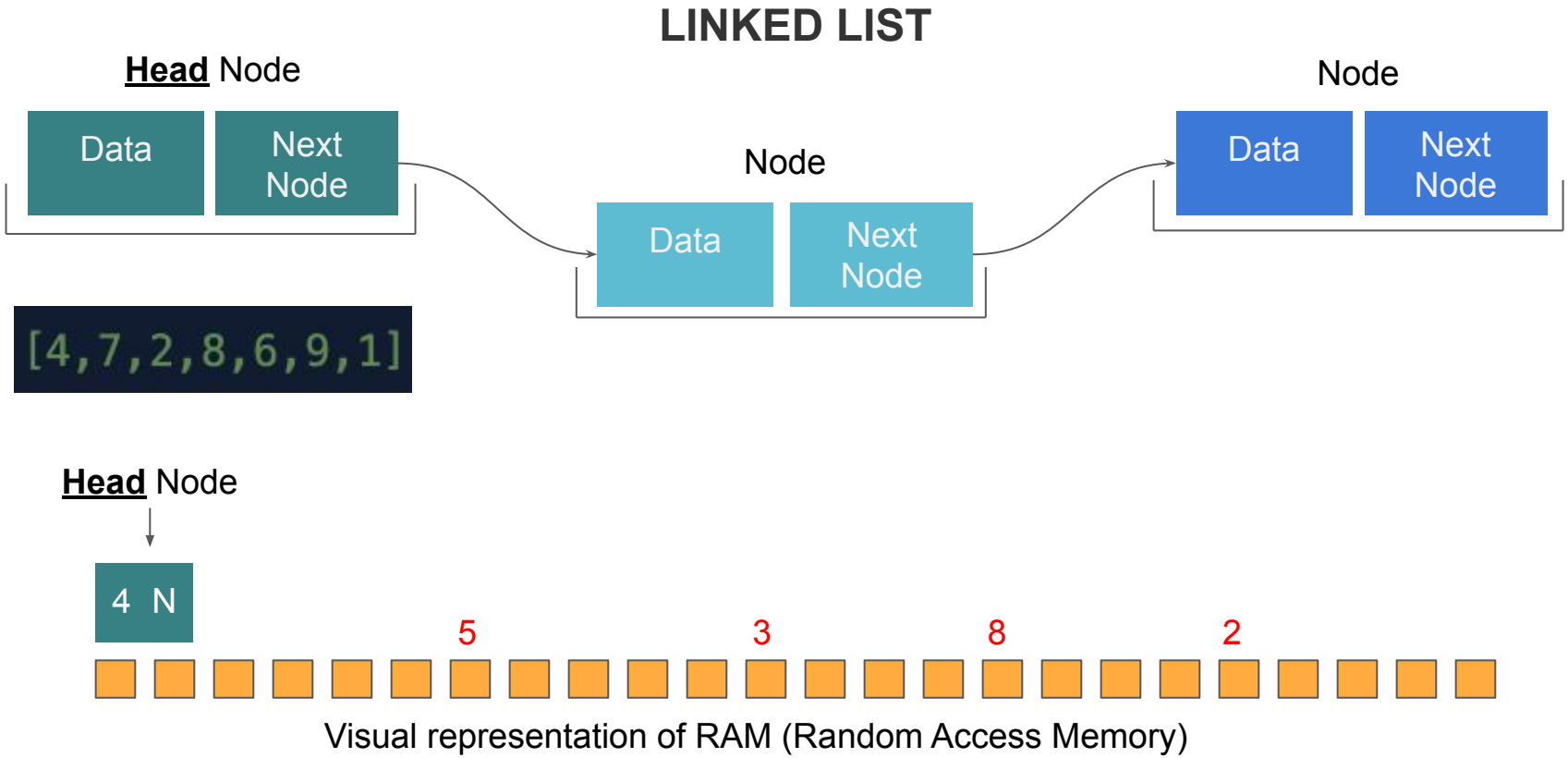
ERROR

[4,7,2,8,6,9,1]

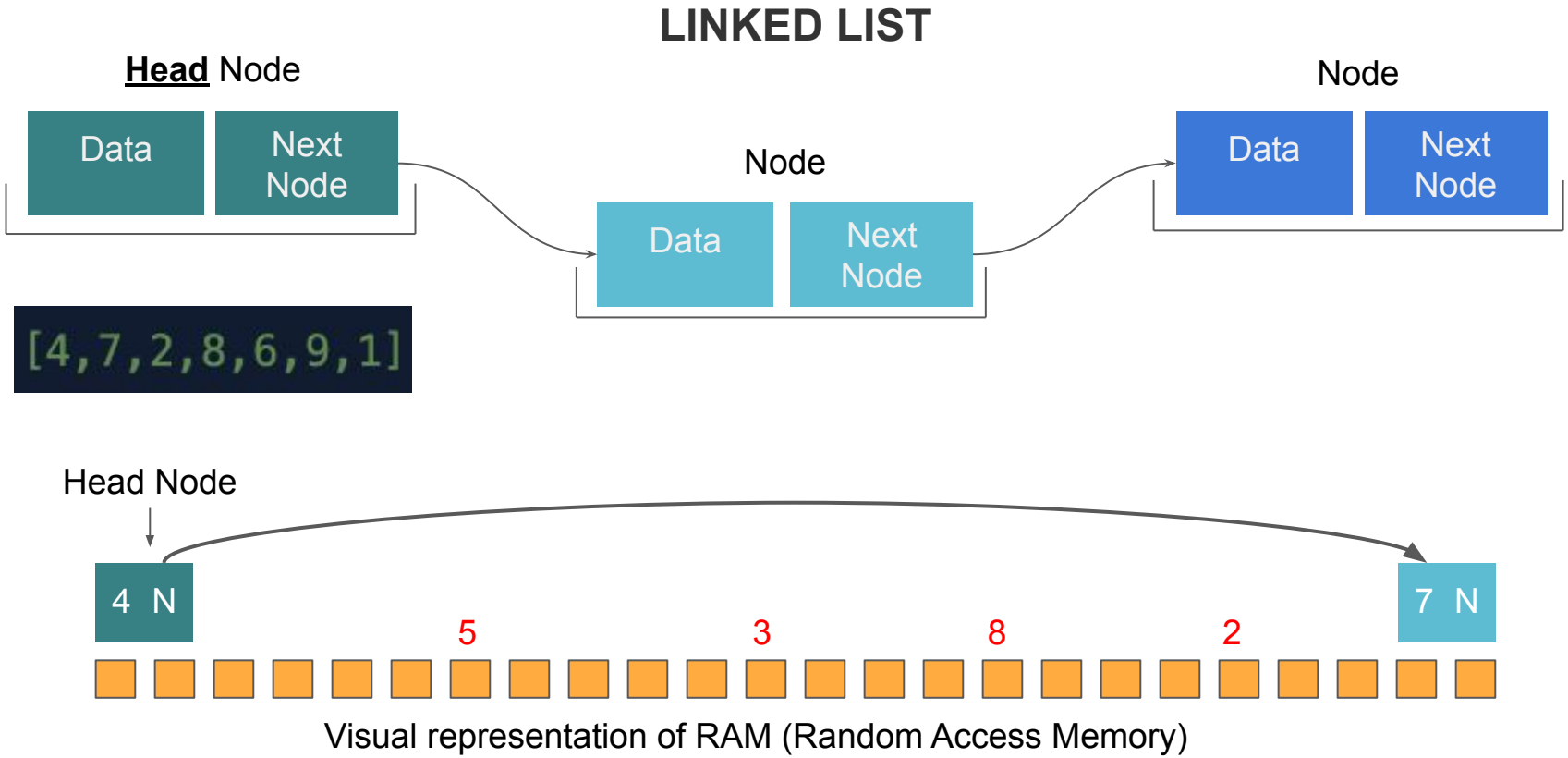


Visual representation of RAM (Random Access Memory)

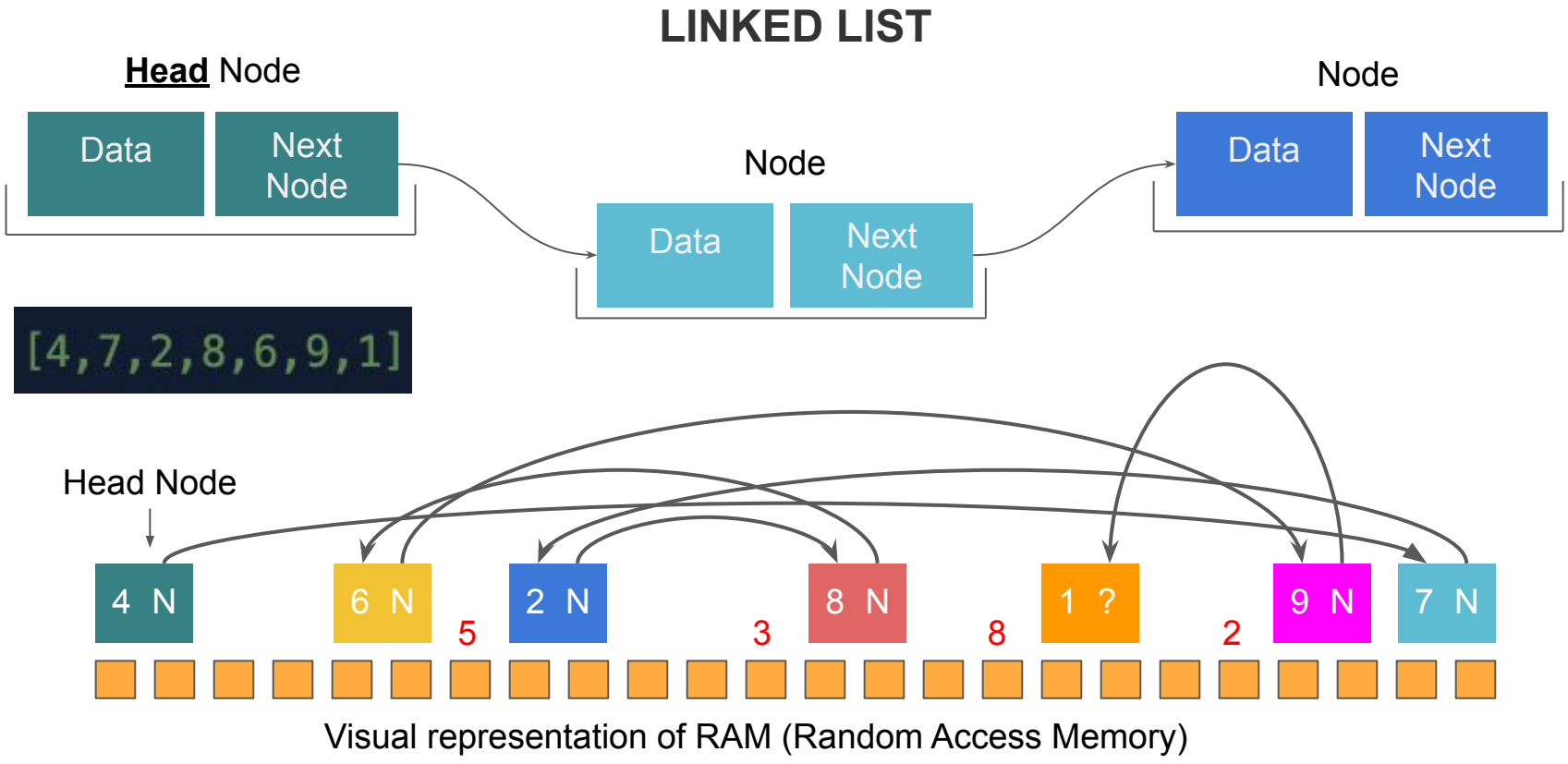
How do we solve this?



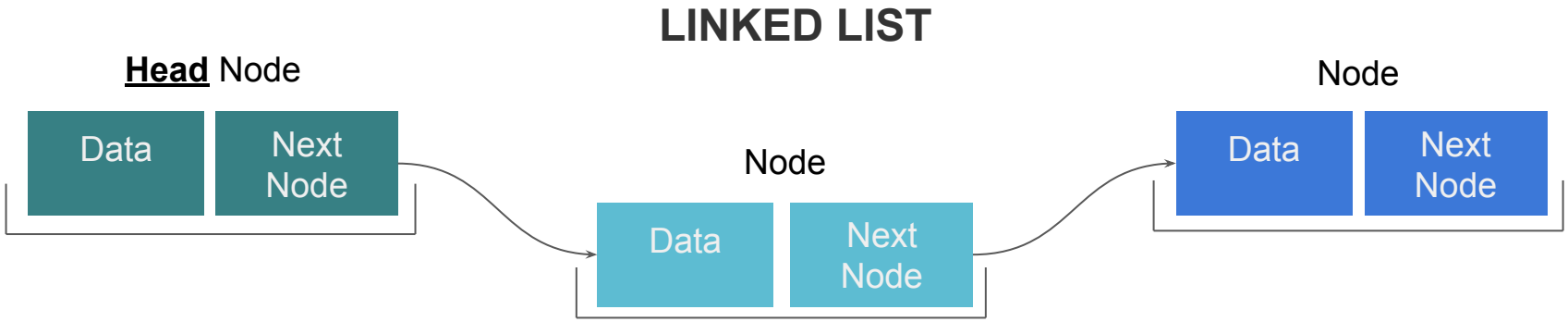
How do we solve this?



How do we solve this?



How do we solve this?



<LET'S START CODING>