Data Structures

Stacks and Queues

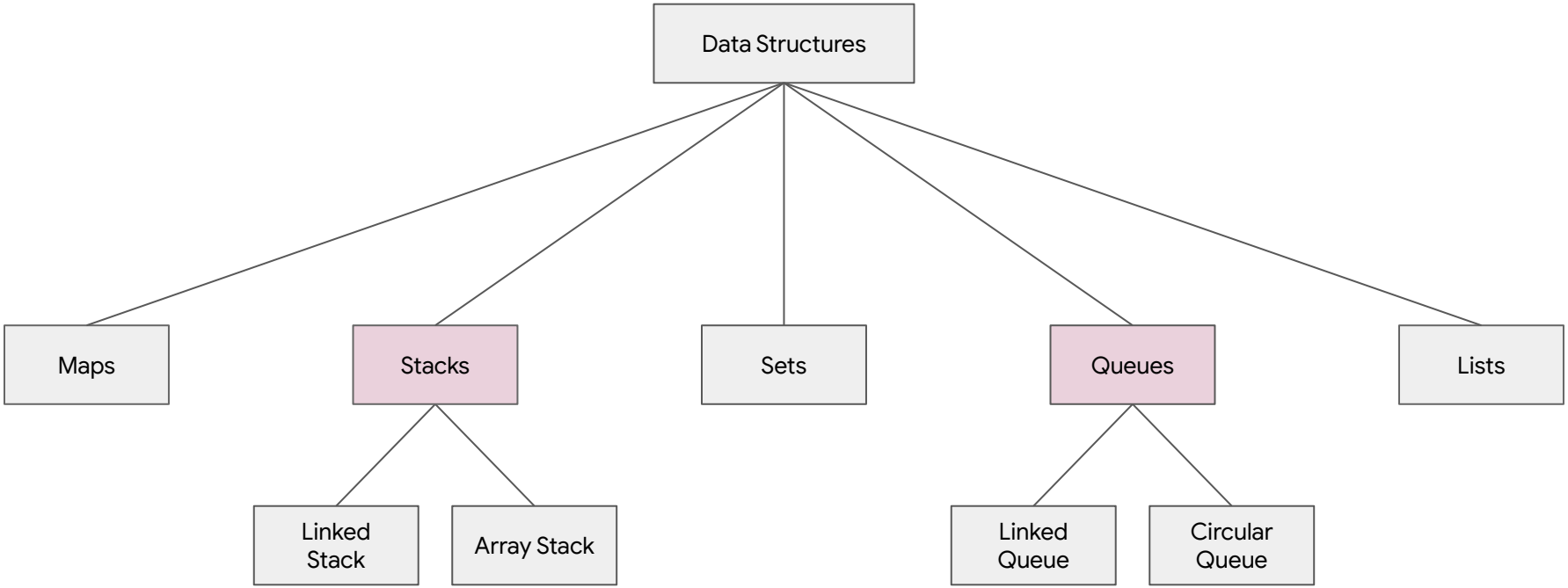
Objectives

Primary Objectives

- What is a **stack** and when could I use it?
- What is a **queue** and when could I use it?

Secondary Objectives

- What trade-offs am I making when I use a **stack**?
- What trade-offs am I making when I use a **queue**?


Data Structures
Maps
Stacks
Sets
Queues
Lists
Linked Stack
Array Stack
Linked Queue
Circular Queue

Stacks

Characteristics

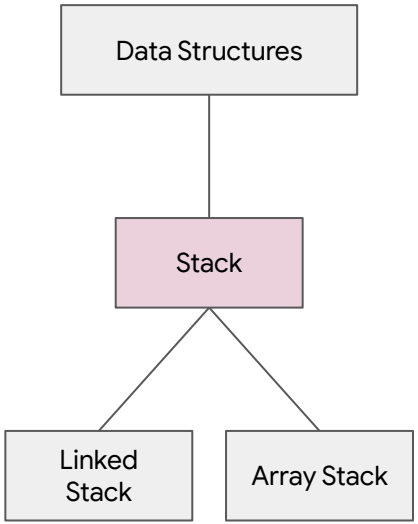
- First-in, last-out

Linked Stack

- Based on the strengths and weaknesses of the linked list.

Array Stack

- Based on the strengths and weaknesses of the linked list.
- Most common implementation.



Stacks

Push (add)

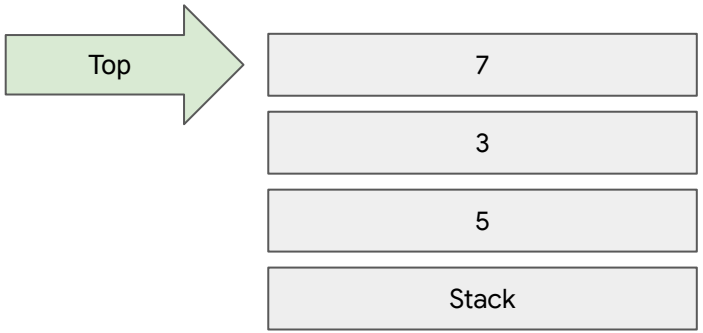
- Add a value to the top of the collection.

Pop (remove)

- Remove a value from the top of the collection.

```
Stack<Integer> s = new Stack<Integer>();
s.add(5);
s.add(3);
s.add(7);

s.remove(); // returns 7
s.remove(); // returns 3
s.remove(); // returns 5
```



Stack Summary

General Strengths and Weaknesses

- Relies on the underlying data structure.
- Differences are often minor and exist at the fringes.

Runtime Analysis	
push	O(1)
pop	O(1)

Queues

Characteristics

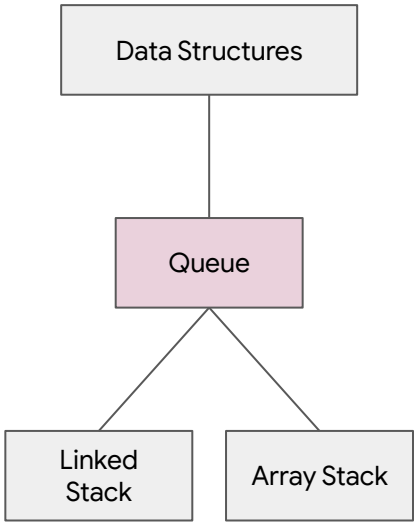
- First-in, First-out

Linked Queue

- Based on the strengths and weaknesses of the linked list.
- Good for unbounded and non-blocking queues.

Array Queue

- Based on the strengths and weaknesses of the linked list.
- Good for bounded and blocking queues.



Queues

Push (add)

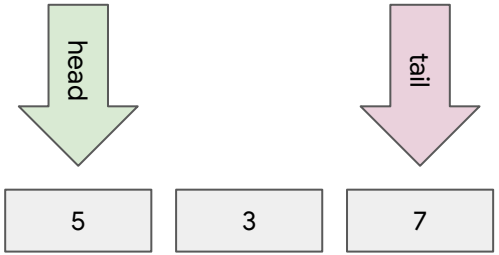
- Add a value at the tail of the collection.

Pop (remove)

- Remove a value from the head of the collection.

```
Queue<Integer> q = new LinkedList<Integer>();
q.add(5);
q.add(3);
q.add(7);

q.remove(); // returns 5
q.remove(); // returns 3
q.remove(); // returns 7
```



Queue Summary

General Strengths and Weaknesses

- Depends on the intended behaviour.
- Implementation will depend on which behaviour you want.

Runtime Analysis	
push	O(1)
pop	O(1)

Performance

Why do stacks and queues have such good runtime performance?

	Stacks	**Queues**
Push	O(1)	O(1)
Pop	O(1)	O(1)

Reflection

Primary Objectives

- What is a **stack** and when could I use it?
- What is a **queue** and when could I use it?

Secondary Objectives

- What trade-offs am I making when I use a **stack**?
- What trade-offs am I making when I use a **queue**?